zmanda
Open Source Backup
zmanda
Recovery Manager
FOR MySQL

Table of contents

Welcome 5
Conventions followed in the Manual 5
Introduction 5
Backup Sets 6
ZMC Components 6
ZMC and the Zmanda Network 6
Features 6
Backup Features 6
Recovery Features 7
Reporting and Scheduling Capabilities 7
Plugins 7
Window Server Differences 7
Tested Platforms 8
ZRM Backup Server Requirements 9
Linux/Solaris 9
Windows 9
ZRM Server Package Dependencies 9
Linux 10
Solaris 10
Windows 11
MySQL server (Backup Client) Requirements 11
MySQL server (backup clients) package dependencies 11
Linux 11
Solaris 12
Windows 12
MySQL database user for ZRM 12
Required Privileges for the MySQL Account 13
Enabling MySQL server binary logs 13
Requirements for Snapshot and Storage Checkpoint Backups 14
Requirements for InnoDB Hot Backup/MySQL Enterprise Backup 14
SSL Between MySQL Servers and ZRM server 14
Downloading ZRM for MySQL Packages 15
ZRM Server Components 16
Installing License Keys 16
Linux 16
Solaris 16
Windows 16
MySQL Server Components 17
Linux 17
Solaris 17
Windows 17
The ZMC Rapid Installer (Linux and Unix) 17

Rapid Installer Command Line Options . . . 18
Uninstalling ZRM for MySQL (Linux and UNIX) . . . 19
The ZMC Rapid Installer (Windows) . . . 19
Uninstalling ZRM for MySQL Windows Server . . . 20
Secure Socket Layer (SSL) Certificate for ZMC web server . . . 20
File Locations on ZRM Server . . . 21
File Locations on MySQL server . . . 21
Logging in to the Zmanda Management Console . . . 22
Initial User name and Password . . . 22
Lost Password . . . 23
Zmanda Network Authentication . . . 23
About Backup Sets . . . 24
What a Backup Set Contains . . . 24
Multiple backup set configurations . . . 25
Backup Set Starter Page . . . 25
Create New Backup Set . . . 26
Backup Set Dashboard . . . 26
Specifying what to Back Up . . . 27
Server Parameters . . . 28
MySQL User Parameters . . . 29
What to Backup . . . 29
What to Exclude . . . 30
Backing up MySQL Application-specific files . . . 30
MySQL cluster (NDB storage engine) backup/recovery . . . 32
Backup What page . . . 33
Backup Where . . . 33
Backup When . . . 35
Backup How . . . 37
Backup Parameters . . . 37
Backup Alerts/Logging . . . 39
MySQL Integration Parameters . . . 40
Plugin Parameters . . . 41
Snapshot Plugins . . . 47
Supported Mechanisms . . . 47
Quick (No-copy) option . . . 48
Setting Retention Policies for the Quick Snapshot Option . . . 48
Linux Logical Volume Manager Snapshots . . . 48
MySQL Configuration . . . 49
Pre-conditions for Using LVM Snapshots . . . 49
Configuration Parameters . . . 50
Advantages of Using LVM Snapshots for Backup . . . 51
Disadvantage of Using LVM Snapshots for Backup . . . 51
Symantec VxFS snapshots . . . 52
MySQL Configuration Requirements . . . 52
Pre-conditions for Using VxFS Snapshots . . . 53

Configuring ZRM for MySQL to use VxFS Snapshots . . . 53
Network Appliance SnapManager . . . 54
MySQL Configuration Requirements . . . 54
Pre-conditions for Using Network Appliance Snapshots . . . 55
Configuring ZRM for MySQL to use ZFS SnapManager . . . 56
ZFS Snapshots . . . 56
MySQL Configuration Requirements . . . 57
MySQL Backup User sudo Privileges . . . 58
Configuring ZRM for MySQL to use ZFS Snapshots . . . 58
EMC SnapView Snapshots . . . 59
MySQL Configuration Requirements . . . 59
Pre-conditions for Using EMC Snapview Snapshots . . . 60
Configuring ZRM for MySQL to use EMC SnapView Snapshots . . . 61
Bluearc Titan Snapshots . . . 61
MySQL Configuration Requirements . . . 62
Pre-conditions for Using Bluearc Titan Snapshots . . . 62
Configuring ZRM for MySQL to use Bluearc Titan Snapshots . . . 63
Amazon EC2 Cloud . . . 64
Requirements . . . 65
Configuring ZRM for MySQL to use Amazon EBS Snapshots . . . 66
Restoring EBS snapshot backup . . . 67
Network Appliance SnapManager . . . 68
MySQL Configuration Requirements . . . 68
Pre-conditions for Using Network Appliance Snapshots . . . 69
Configuring ZRM for MySQL to use ZFS SnapManager . . . 70
Monitoring Backups . . . 71
Backup Status icons . . . 72
Events . . . 72
Event Viewer . . . 73
Filter Events . . . 74
Log Rotate Utility . . . 74
Summary Report . . . 74
Selecting a backup date . . . 75
Legend . . . 76
Summary Panel . . . 76

Report Custom . . . 77

Custom reports . . . 77
Predefined Reports . . . 78
Backup Report . . . 79
Application Impact Report . . . 79
Backup Status Report . . . 80
Backup Method Report . . . 81
Backup Retention Policy Report . . . 82
Backup Performance Report . . . 83

Incremental Backup Report . . . 84
Replication Backup Report . . . 85
Cluster Backup Report . . . 86
Customized Report . . . 87
Database events report . . . 87
Selecting Backup Date . . . 89
Database Event Viewer . . . 90
Database Event Search . . . 90
Launching Restores from the Database Events Reports . . . 92
Backup image integrity report . . . 93
Convert Quick backups to Standard backups . . . 94
ZMC User administration . . . 95
ZMC User . . . 95
Associating with Zmanda Network . . . 96
Editing/Deleting ZMC User . . . 96
Admin Backup sets . . . 97
Managing Backup Sets . . . 97
Backup Set Creation . . . 97
Modifying Backup Sets . . . 98
Set Site Defaults . . . 98
Admin Preferences . . . 99
ZMC services status . . . 100
ZRM License status . . . 101
Restore What . . . 102
Restore From . . . 103
What to Restore . . . 103
Restore Where . . . 105
EBS snapshot restore parameters . . . 106
Restore How . . . 107
Copy Plugin Parameters . . . 107
Restoring to a MySQL replication slave . . . 110
Restoring a table from a database backup . . . 110
Notes on Browsers . . . 111
The Browser "Remember Input" Option . . . 111
Page Refresh Considerations . . . 111
Using the Zmanda Troubleshooting Scripts . . . 111
Running the Support Script on the ZRM server . . . 111
zm-support Options . . . 112
Running the Windows Client Support Script . . . 113
Location . . . 113
Files Gathered . . . 113
Zmanda Client for Windows Installation Logs . . . 113
Zmanda Client for Windows Debug Logs . . . 113
System Logs . . . 113

Additional information 114
Output File 114
Error Messages 114
Failure Messages 114
Warning Messages 119

Welcome to the Zmanda Management Console for ZRM for MySQL Enterprise Edition 3.3 User Guide. This manual describes how to use Zmanda Management Console web based user interface for managing backups on ZRM Enterprise Servers.

- Installing Zmanda Management Console.
- Configuring ZMC Backup Sets.
- Managing ZMC backup Sets.

Conventions followed in the Manual

- Command line examples and program output are displayed in a box:

```
# This is a system message.
```

- Menu options, button names, and other interactive controls are shown in **boldface** type. For example: Choose **Configure...** from the pop-up menu.
- Boldface type also denotes literal portions of a command line; italic type denotes a variable that you must substitute with an actual value. For example,
  Enter **copy** *filename*

means that you should type "copy" followed by the actual file name you wish to copy, followed by the Enter key.

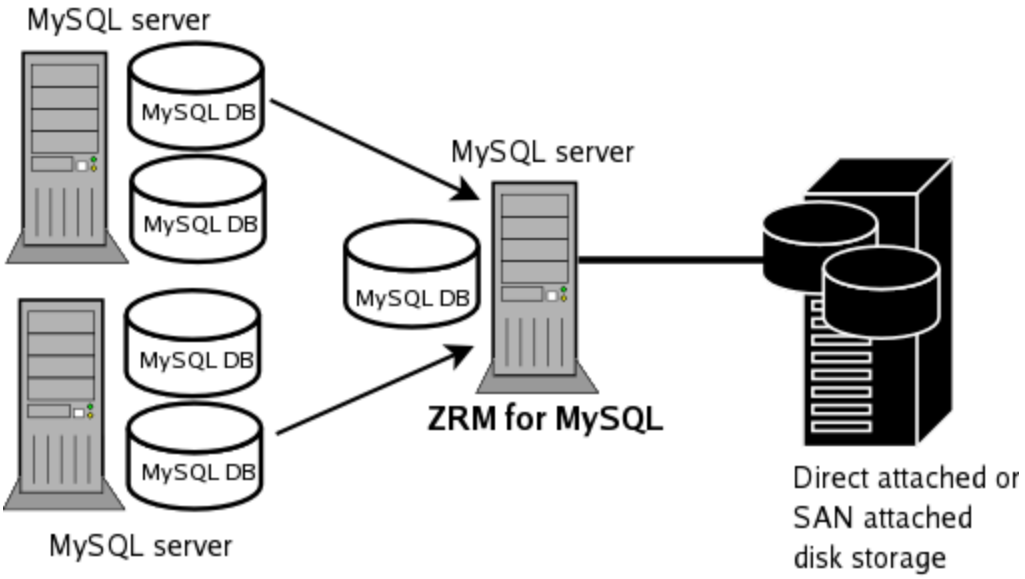
- The term **Localhost** is used to refer to the server on which ZMC is installed and the term **Remote host** is used to refer to other servers that are to be backed up.

Introduction

MySQL is one of most common and versatile database engines. ZRM Enterprise provides global backup and recovery management for MySQL servers.

In addition to its command line and configuration file interface, ZRM for MySQL also includes the Zmanda Management Console for MySQL (ZMC). The ZMC is a a web-based management console that delivers all the automation, management, monitoring, and recovery capabilities of ZRM for MySQL in an intuitive graphical user interface.

The following diagram shows the local server running ZMC backing up two more MySQL servers with multiple MySQL databases. These are backed up by configuring at least two backup sets i.e. at least one backup set per MySQL server.



Backup Sets

Most ZMC operations work on backup sets. A backup set defines of the "what", "where", "when" and "how" of data that is to be backed up. ZMC users can have two roles: Administrator or Operator. All backups and reports previously created using community versions of ''ZRM'' can be managed using ZMC.

ZMC Components

ZMC consists of the following components:

- Apache web server
- Internal MySQL database server
- PHP, Perl and PHP/Perl modules.

All ZMC components are installed in a separate directory (under **/opt/zmanda/zrm** and **/opt/zmanda/common**), and therefore do not impact existing installations of AMP stack on the backup server.
ZMC runs on the ZRM server and can be accessed from any web browser. ZMC is supported on IE 8 and IE 9 on Windows, Firefox 3.5 or greater on Windows and Linux.

ZMC and the Zmanda Network

The ZMC is closely integrated with the Zmanda Network, which provides documentation as well as context-sensitive help on messages and error resolution. The Zmanda Network is continuously updated with the latest ZRM information. This allows the ZMC to provide up-to-date suggestions and resolutions for specific issues faced by a backup administrator.

Features

ZRM for MySQL optimizes backup and restore operations on MySQL Databases. It provides full flexibility to individually leverage native MySQL/OS backup tools, levels, scheduling etc. It generates logs that assist in optimizing these capabilities. It also provides filters to easily locate anomalous database events.

Backup Features

- Backs up multiple MySQL databases managed by one or more MySQL servers.
- Backs up tables in a single database.
- Supports hot backup of databases.
- Supports multiple backup methods, depending on the storage engine used by MySQL tables.
- Support for Xtrabackup and MySQL Enterprise backup tools.
- Full, differential, and incremental database backups.
- Supports use of mysqldump, mysqlhotcopy, lvm snapshots and MySQL replication to execute backups.
- Creates consistent backups of the database regardless of the storage engine used by database tables.
- Supports SSL authentication between the local ZRM for MySQL and remote MySQL server to allow secure backups over the Internet and across firewalls.
- Verifies backed up data images.
- Backup images can be compressed as well as encrypted using standard tools such as *gzip*, *GPG*, etc .
- Backup runs can be cancelled by users with Administrative privileges.

Recovery Features

- Optionally maintain a backup index that stores information about each backup run.
- Includes a reporting feature that can be used to browse indexes.
- Supports recovery of full and incremental database backups.
- Perform selective, incremental restores based on binary log position, or by a given point in time, thus protecting the database from operator errors.
- Binary log filtering helps you decide what to restore and what to discard.
- Depending on the type of backups selected, data can be recovered to the same machine or to a different machine.

Reporting and Scheduling Capabilities

- ZRM for MySQL can schedule backup runs immediately or in daily/weekly/monthly intervals.
- It automatically generates backup reports.
- It includes standard backup reports as well as custom (i.e. user-defined) backup reports.
- It can report any backup statistic or combination thereof.
- It can format reports as HTML or Text .
- It sends email notification about the backup run status.
- It can provide backup reports as RSS feed.

Plugins

ZRM for MySQL allows *plugin* operations to extend ZRM for MySQL capabilities. Plugins allow you to optimize the backup process for your environment. The following plugin operations are supported (see the Backup How page for detailed configuration information):

- Pre-scheduling
- Pre-backup
- Post-backup
- Copy plugins for Linux/Unix and Windows
- Binary log parser plugin
- Snapshot plugin (feature license required)
- InnoDB Hot Backup plugin (feature license required)

Window Server Differences

When running Windows-based ZRM servers (XP, 2003 Server, Vista), the following limitations apply:

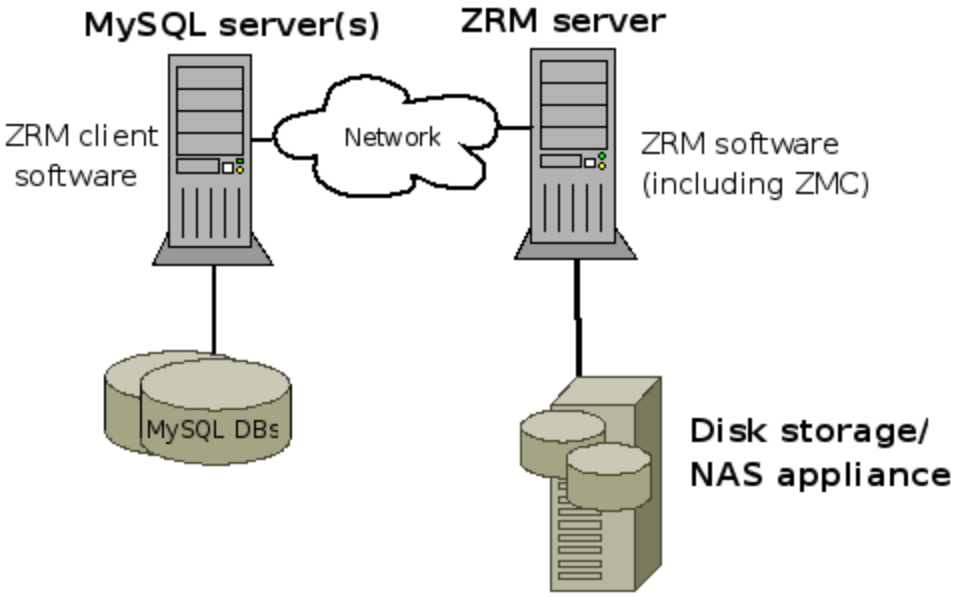
- ZRM for MySQL 3.3 is not available for Windows. ZRM 3.1 Windows server and ZRM 3.2 Windows clients works with ZRM for MySQL 3.3.
- Backup methods that are supported are logical and snapshot backup using VSS. Raw backup method is not supported by MySQL on Windows
- No support for backup of Linux/Solaris MySQL servers; use ZRM on Linux or Solaris to back up MySQL servers running on multiple platforms to a single ZRM server.
- Support for native Windows compression & encryption. No support for encryption on Vista Home basic and Vista Home Premium.
- Backup of replication slaves running on Windows
- No Quick Snapshots. The number of snapshots that can be maintained on XP is limited to 1
- No backup data verification
- No support for MySQL clusters.
- No email/RSS notifications. Information will be logged in Windows event logs

Tested Platforms

Please see the latest platform support information on the Zmanda Network.

This page provides the list of requirements for ZRM Backup Server, ZRM client (MySQL server), MySQL server configuration and requirements for various backup methods. Users must read this section before starting to install ZRM for MySQL components as described in Downloading and Installing instructions section.

Throughout this document, The *MySQL Server* refers the database server being backed up by the *ZRM for MySQL Server,* which also called the *ZRM Server.*



ZRM Backup Server Requirements

- ZRM installer and the ZRM Server requires 256MB of free memory. You should have at least 500MB of disk space, and you should not install the ZMC (**/opt/zmanda/zrm**) on an NFS or a network partition.
- Ensure that there is at least 10% free disk space in the Zmanda installation (**/opt/zmanda/zrm**) and temporary directories (**/tmp**).

Linux/Solaris

- Verify that the ZRM server has operating system accounts for a **mysql** user belonging to the **mysql** group. The account should have a login shell, and be able to execute commands on the ZRM server. If implementing a copy plugin to back up clients, matching accounts must be set up on the MySQL servers. The **uid** and **gid** of the **mysql** user must match on the ZRM server and the MySQL server (backup client).
- The **mysql** user must be added to **cron.allow** file. This file is usually in **/etc** or **/etc/cron.d** directory. The crontab file must be also editable by **mysql** user.
- On ubuntu platform, the **crontab** for **mysql** must set the **MAILTO** parameter to a valid email address (or an empty string). The following example shows the entry for setting the parameter to an empty string:

```
MAILTO=""
```

- The ZRM server must have the locale set to **en_US.UTF-8** (US English, UTF-8 Character set).

Windows

- ZRM server must be installed and run from the Administrator account.
- Windows server must be running US English edition

ZRM Server Package Dependencies

The following packages are required on the ZRM Server (32-bit versions of packages required):

Linux

- ZRM for MySQL requires the MySQL client commands listed below to be installed on the ZRM server. ZRM uses the following MySQL commands on the ZRM server:
  - ? **mysqladmin**
  - ? **mysqlhotcopy**
  - ? **mysqldump**
  - ? **mysqlbinlog**
  - ? **mysql**
- The MySQL client commands installed on the ZRM server must be compatible with the version of MySQL servers being backed up. Zmanda recommends installing the same version of MySQL software on the ZRM server and MySQL server. The ZRM server uses *mysqlbinlog* command to parse binary logs created by the MySQL servers. So, the version of *mysqlbinlog* command must be compatible with binary logs generated by different MySQL servers that are being backed up.
- The ZRM server assumes that the MySQL commands are installed in **/usr/bin/**. You can change this default either globally or by backup set. If necessary, you can change the client command location and binary log location in Set Site Defaults page. See the MySQL subsection of the Backup Tab for details.
- On some distributions (such as debian), the default mysql binary log location */var/log/mysql* is owned by **mysql** user and **adm** group. ZRM expects the binary logs to be owned by **mysql** user and **mysql** group. Please change the group ownership of the directory.
- **mailx**: The **mailx** package can be found as part of the Linux distribution. The **mailx** package must be configured to send mail from the ZRM machine to the MySQL database backup administrator.
- On 64-bit ubuntu/debian platforms, ZRM installer needs *ia32-libs*.

The dependency packages are installed by default on most Linux distributions. If you need to add them, you can use **yum** or **apt-get**, either from the distribution media, or from a distribution repository (run either as root):

```
#yum install package_name
```

or

```
#apt-get install package_name
```

Solaris

- ZRM for MySQL requires the MySQL client commands listed below to be installed on the ZRM server. ZRM uses the following MySQL commands on the ZRM server:
  - ? **mysqladmin**
  - ? **mysqlhotcopy**
  - ? **mysqldump**
  - ? **mysqlbinlog**
  - ? **mysql**

- The MySQL client commands installed on the ZRM server must be compatible with the version of MySQL servers being backed up. Zmanda recommends installing the same version of MySQL software on the ZRM server and MySQL server.
- The ZRM server assumes that the MySQL backup client's MySQL commands are installed in **/usr/bin/**. You can change this default either globally or by backup set. If necessary, you can change the client command location and binary log location in Set Site Defaults page. See the MySQL subsection of the Backup Tab for details.
- **mailx**: The **mailx** package can be found as part SUNWcsu package. ZRM uses **mailx** program to send email notification.
- GNU utilities (such as GNU tar, GNU find) are also required on the server.

The Zmanda Network downloads page collects all these packages in a downloadable tar archive (which is made available after you select a platform to download). If necessary, download the tar archive to the local system, extract it, and run **pkgadd** as the superuser to install any necessary packages:

```
# pkgadd -d package_name
```

Windows

- ZRM server can only run on 32bit Windows platform.
- ActivePerl version 5.8.8 is required. Please contact Zmanda if you do not have access to ActivePerl version 5.8.8. After downloading the file, run the **Setup** program. After you confirm the license agreement, the **Setup** program asks you what programs to install and prompts you for some installation details. Install all of the products in their default locations, and make sure that the **Add Perl to the PATH environment variable** and **Create Perl file extension association** boxes are checked before clicking through the **Finish** dialog.
- Additional Perl modules are also required. These are installed as part of ZRM installation. ZRM installer requires connection to Internet during installation.

MySQL server (Backup Client) Requirements

- Verify that the version(s) of MySQL server(s) that you deploy are supported by Zmanda Recovery Manager for MySQL. Check the supported platform compatibility matrix.
- You must set up an operating system account for the **mysql** user. Our recommendation is the mysql user should have UID and GID that match mysql user on the ZRM server especially for ssh copy plugin. This user must have read/write access to the MySQL data directory and binary logs.
- MySQL Enterprise Backup or Xtrabackup (1.6 or later) tools must be installed on the MySQL server if the tool is going to be used for full backups.

MySQL server (backup clients) package dependencies

Linux

- **perl:** The **perl-DBI**, **perl-DBD-MySQL**, and **perl-XML-Parser** modules are also required.
- **perl** LWP-UserAgent module is required for Netapp snapshot backups.
- **xinetd**
- **sudo** (required only for snapshot backup configurations)
- Gnu **tar** version 1.15 or later.

These programs are installed by default on most Linux distributions. If you need to add them, you can use **yum** or **apt-get** to install them, either from the distribution media, or from a distribution repository (you run either as root):

```
#yum install package_name
```

or

```
#apt-get install package_name
```

Solaris

- **perl**: The **Perl-DBI**, **Perl-DBD-MySQL**, and **perl-XML-Parser** modules are also required.
- **perl** LWP-UserAgent module is required for Netapp snapshot backups.
- **SUNWcsu**
- **SUNWsshcu**
- **SUNWgzip**
- **SMCtar** (version 1.15.1 or greater)
- **SMCgpgme**
- **SMCsudo** (required only for snapshot backup configurations)
- **SMCgrep**
- **SMCcoreu**

The Zmanda Network downloads page collects all these packages in a downloadable tar archive (which is made available after you select Solaris platform). If necessary, download the tar archive to the local system, extract it, and run **pkgadd** as the superuser to install any necessary packages:

```
# pkgadd -d package_name
```

If you are using ZRM dependency tar ball for installation, you may need additional dependency common-1.4.5-SunOS5.8-sparc-CSW.pkg that is not part of the tar ball.

Windows

- Databases to be backed up must be stored on NTFS volumes with the Volume Shadow Copy Service (VSS) enabled.
- Windows clients must open inbound TCP ports 10080 and 10081, and outbound TCP ports 700:800.
- Windows clients must open inbound TCP ports 10080 and 10081, and outbound TCP ports 700:800.
- The Remote Registry Service must be enabled before installation.

MySQL database user for ZRM

We recommend creating a MySQL database user for ZRM backup and recovery instead of using the MySQL root user. If the MySQL database backup user and restore user are different, set the privileges of the backup user in **Backup Tab** for the backup set.

Restore user privileges can be specified on the **mysql-zrm(1)** command line via the **--user** and **--password** options.

Although MySQL allows dollar signs (**$**) in passwords, the ZMC does not. If you must use passwords that include dollar signs at your site, consider using one of the alternative methods for storing MySQL user passwords described in the MySQL article "Keeping your Password Secure".

Required Privileges for the MySQL Account

MySQL backup and restore users need, at minimum, the following MySQL privileges:

backup user
LOCK TABLES, SELECT, FILE, RELOAD, SUPER, UPDATE, TRIGGER, SHOW VIEW, EXECUTE, EVENT

restore user
CREATE, DROP, INDEX, SHUTDOWN, INSERT, ALTER, DELETE, UPDATE, TRIGGER, SUPER, REPLICATION CLIENT, CREATE VIEW

```
EXECUTE, SHOW EVENTS, SHOW VIEW and CREATE VIEW privileges are required for MySQ
5.1.6, so for prior versions of MySQL remove those options from the list above.
create or drop triggers.
```

MySQL backup user requires SUPER privileges even when MySQL replication is not being used. For incremental backups, ZRM for MySQL requires SUPER privileges to enable binary logging.

A MySQL replication slave backup user should have REPLICATION CLIENT privileges in addition to the above privileges.

Example: Command that grants minimal user privileges for database user **dbabackup** to backup and restore database **expenses** (MySQL 5.1 server) remotely from machine **server.company.com** :

```
mysql> GRANT LOCK TABLES, EXECUTE, SELECT, FILE, RELOAD, SUPER, CREATE, DROP, IN
    -> ON expenses.*
    -> TO 'dbabackup'@'server.company.com'
    -> IDENTIFIED BY 'obscure';
```

ZRM for MySQL should be running on **server.company.com**. If you are restoring from logical backups, additional privileges will be required for the restore user. For example, if there are stored procedures in the logical backup image being restored, the restore user must have **CREATE ROUTINE** and **ALTER ROUTINE** privileges. If you are not sure of the list of privileges that are required for restoration, temporarily grant the restore user all privileges for the databases and/or tables being restored.

Note: If you are backing up remote MySQL server, backup user privileges should be granted for the backup user accessing from the MySQL server as well as the server running ZRM (ZRM server).

Enabling MySQL server binary logs

MySQL backups (full and log incremental) require that binary logging on the MySQL server. To enable binary logging, start the MySQL server daemon (**mysqld**) with the **--log-bin** option:

```
mysqld --log-bin=BinLogFilename
```

Enabling binary logs on a MySQL server reduces performance by about 1%. Actual performance degradation depends on the type of database workload. It is the best practice to store binary logs in a different file system than the file system containing the database directories.

Consult MySQL reference manual for more information on MySQL binary logs.

log-slave-updates must be enabled on the replication slave mysqld options file (my.cnf) if you are performing backups of a MySQL replication slave. Please note when replication master is switched over to a slave and there is chained replication (master -> slave -> slave), there might be duplicate events if **log-slave-updates** is enabled on the slave.

Requirements for Snapshot and Storage Checkpoint Backups

ZRM for MySQL can create temporary snapshots of the file systems or storage volumes and use these snapshot volumes to do backups. If the database resides on a Veritas File System (VxFS), storage checkpoints can be leveraged.

Various storage and filesystem snapshots are supported. Some snapshot mechanisms are licensed and will require licenses to be purchased. All MySQL database files (data, log, indexes) must be stored on snapshot-capable storage volumes.

The requirements for each file system and storage snapshot is different. They are discussed in the Snapshot Plugins chapter.

Requirements for InnoDB Hot Backup/MySQL Enterprise Backup

The **ibbackup** command (which is installed as part of the MySQL Enterprise Backup or InnoDB Hot Backup product) must be installed on the MySQL Server. MySQL Enterprise Backup or InnoDB Hot Backup product must be purchased from Oracle or at www.innodb.com. ZRM provides integration with MySQL Enterprise Backup or InnoDB Hot Backup product. Make sure the **mysql** operating system user has permissions to execute the command. If **ibbackup** command is installed in a place other than **/usr/bin**, you must specify the path when you choose **InnoDB Hot Backup** on the **Backup How** page.

SSL Between MySQL Servers and ZRM server

SSL provides an additional layer of security while moving backups over a network. We recommended that you enable SSL on the MySQL server if the backups are performed on unsecured networks. Installing SSL between the local ZRM for MySQL server and remote MySQL server(s) is necessary only for logical backups of remote MySQL servers.

To verify the availability of SSL support in the MySQL server, you can either:

- Run the following command on the local server:

```
# mysqld --ssl --help
```

When the SSL support does not exist, the system responds with a message like this:

```
060828 15:25:08 [ERROR] mysqld: unknown option '--ssl'
```

- Examine the value of the **have_openssl** system variable:

```
mysql> SHOW VARIABLES LIKE 'have_openssl';

+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| have_openssl  | YES   |
+---------------+-------+
```

Consult the MySQL reference manual for configuring SSL on MySQL.

Zmanda recommends using either of the two options given below to configure SSL when remote backups of MySQL servers done using unsecured networks.

- Set SSL parameters in the **my.cnf** file of MySQL on the ZRM server.

ssl-ca=*mySQL_conf_dir***/openssl/cacert.pem**
ssl-cert=*mySQL_conf_dir***/openssl/client-cert.pem**
ssl-key=*mySQL_conf_dir***/openssl/client-key.pem**

- Set SSL parameters for all backup set in mysql-zrm.conf file of *ZRM for MySQL* or you can specify it in the **SSL options** field in **Backup What** page of ZMC.

ssl-options="--ssl --ssl-ca=*mySQL_conf_dir***/openssl/cacert.pem**
--ssl-cert=*mySQL_conf_dir***/openssl/client-cert.pem**
--ssl-key=*mySQL_conf_dir***/openssl/client-key.pem"**

Download distribution files from from the Zmanda Network Downloads page. Note that there are a number of different packages for various MySQL server and ZRM server architectures.

If you are installing ZRM for MySQL version 3.3 over a previous version, please see the "Compatibility with Previous Versions of the Zmanda Recovery Manager for MySQL" section of the ZRM for MySQL 3.3 Release Notes.

ZRM Server Components

The *ZRM Server* is the backup server. It can run either Linux, Solaris, or Windows. The **installer.bin** (or, for Windows, **installer-win.exe**) packages are binary executables for the ZMC Rapid Installer, which install the ZRM Server software, the Zmanda Management Console (ZMC) and all dependencies: this is the recommended method of installation. The **installer.bin** packages include the ZRM command line interface and ZRM client software. The installation instructions are provided in the next section.

Installing License Keys

After you have purchased a base license and any feature licenses (such as a snapshot license), the Zmanda Network Downloads page will include an option to download a license key (**zmanda_license**). On Linux/Solaris ZRM servers, download the license file to the **/etc/zmanda** directory, then make sure that the file permissions are set to 644 and that the owner is **root**. On Windows ZRM servers, download the file to *ZRM_installation_dir***etc****zmanda** as the Administrator.

Linux

ZRM-enterprise-3.3-installer.bin
: All supported linux distributions.

Solaris

ZRM-enterprise-3.3-installer-sparc.bin
: Solaris on Sparc platform

ZRM-enterprise-3.3-installer-intel.bin
: Solaris on Intel/AMD platform

Windows

ZRM-enterprise-3.1-installer-win.exe
: Windows on x86 (32bit only)

MySQL Server Components

The *MySQL server* is the machine being backed up (i.e., the backup client). No ZRM client components are required if logical backup and recovery of MySQL server.

Linux

MySQL-zrm-enterprise-client-3.3-1.noarch.rpm

RHEL, SLES and Fedora distribution. Install using the following command (run as root):

```
# rpm -i MySQL-zrm-enterprise-client-3.3-1.noarch.rpm
```

mysql-zrm-enterprise-client_3.3_all.deb

Ubuntu, Debian distribution. Install using the following command (run as root):

```
# dpkg -i mysql-zrm-enterprise-client_3.3_all.deb
```

Solaris

MySQL-zrm-client-3.3-solaris.pkg

Open Solaris and Solaris 10 on Intel/AMD and Sparc platforms. After downloading the Solaris client package, install using the following command (run as superuser) in the directory where the package was downloaded :

```
# pkgadd -G -d .
```

The above command installs the package in all zones.

Windows

ZRM-Windows-*.zip

Windows 32bit and 64bit platforms. Latest version of Windows client is 3.2. Install by extracting the archive and running the resulting **Setup.exe**. Administrator privileges are required for installation.

During installation, you have to specify the name of ZRM server or its IP address. If this information is not available at the time of ZRM client installation, You can use ZRM client configuration utility (**Start > Programs > ZRM Client For MySQL > ZRM Client Configuration Utility > Server**) to set the correct IP address of ZRM server and restart ZWC Service from the Services menu. ZRM client must be registered with ZRM server before backups are performed.

The ZMC Rapid Installer (Linux and Unix)

For installation using a method other than the Rapid Installer, please see the sections that follow.
1. Copy the Rapid Installer binary file to the host where the given component will be installed.
2. Log in to the host as the superuser.
3. Make sure that the Rapid Installer binary file is executable. For example:

```
# chmod +x ZRM-enterprise-3.3-installer.bin
```

4. Run the installer by double-clicking on it, or enter the following command line:

```
#  ./ZRM-enterprise-3.3-installer.bin
```

5. The Rapid Installer then starts. Follow the on-screen instructions.

Important Note: When prompted to choose the **Zmanda Web Server protocol**, we strongly recommend that you choose **https** for security reasons. Even if you choose **http** for browser/ZMC communication, the ZMC still requires HTTPS for internal communication purposes, and will therefore prompt you for an SSL port during installation in all cases.

Note: The installer performs several tasks after creating and populating the Zmanda directories. These are completed after the progress bar (which only tracks the archive extraction) shows 100% completion. These tasks take time. Please wait till they complete.

6. After the ZRM for MySQL binaries have been extracted and installed, the Zmanda Management Console is launched and the **readme** file is displayed. The **readme** file includes the default ZRM for MySQL username and password. You can now login to the console using any supported browser and begin backing up MySQL databases.

Rapid Installer Command Line Options

Run the installer with the **--help** option to see what command line parameters are available.

--help
Display the list of valid options.

--version
Display product version information.

--optionfile *optionfile*
Use command line parameters specified in *optionfile.*

--mode *mode*
Choose the installation mode, where *mode* is **gtk** (the default), **xwindow**, **text**, or **unattended**.

--apache_server_port ***apache_server_port***
The ZMC Web Server port (default is **80**).

--apache_server_sslport ***apache_server_sslport***
ZMC Web Server SSL port (default is **443**).

--mysql_port ***mysql_port***
Specify the ZMC MySQL Server port (default is **3036**).

Uninstalling ZRM for MySQL (Linux and UNIX)

Important Note to Customers of Amanda Enterprise Edition: Before running the uninstall script for ZRM for MySQL, you must first stop the Amanda Enterprise Edition services from running (enter **/etc/init.d/zmc stop** as root), then restart it manually after the uninstall script completes (**/etc/init.d/zmc start**).

You can unistall ZRM for MySQL by clicking the uninstall script located in **/opt/zmanda/zrm/uninstall**. Using this script, you can remove the ZRM for MySQL binaries, with the option of leaving configuration files intact. Follow the on-screen instructions after running the script.

The ZMC Rapid Installer (Windows)

Installation must be run from the Administrator account. First, make sure that system meets the requirements for ZRM servers on Windows. If the correct version of ActivePerl is not installed (5.8.8) before installing ZRM for MySQL server, the ZRM installation program will prompt you to install it before it will allow you to continue. If you do not have access to this version of Perl, please contact Zmanda Support Team.

1. Obtain the Windows ZRM server installation program **ZRM-enterprise-3.1-installer-win.exe** from the Zmanda Network Downloads page. Go to the folder where you downloaded the file and double-click it to start the installation. Stay connected to the internet during the installation process. Installing some of the dependencies requires internet access.
2. Follow the on-screen instructions. Because the installation program installs the Zmanda Windows Client as well as the web server and ZRM server, it must extract a large number of files before the setup process can begin. Please be patient.
3. **Important Note**: When prompted to choose the **Zmanda Web Server protocol**, we strongly recommend that you choose **https** for security reasons. Even if you choose **http** for browser/ZMC communication, the ZMC still requires HTTPS for internal communication purposes, and will therefore prompt you for an SSL port during installation in all cases.

The installer performs several tasks after creating and populating the Zmanda directories. These are completed after the progress bar (which only tracks the archive extraction) shows 100% completion. These tasks take time. Please wait until they complete.

After the server and client software has been extracted and installed, the Zmanda Management Console is launched and the **readme** file is displayed. The **readme** file includes the default ZRM for MySQL username and password. You can now login to the console using any supported browser and begin backing up MySQL databases.

The following services are installed as part of ZRM for MySQL server, and must be running for the ZRM server to function:

- ZMC Service
- ZRMEnterpriseApache
- ZRMEnterpriseMySQL
- ZWC service

Uninstalling ZRM for MySQL Windows Server

You can use the **Uninstall ZRM** option added to the Windows **Start->All Programs** menu or the **Add/Remove Programs** option from the Control Panel to remove ZRM for MySQL from the system. After you initiate the uninstall, you are prompted whether you would like to remove backup configuration data as well as the program itself. If you plan to upgrade, you should choose to keep the configuration data.

Note that removing the ZRM Windows Client requires that you use the **Add/Remove Programs** option available on the Windows Control Panel.

Secure Socket Layer (SSL) Certificate for ZMC web server

Although ZRM for MySQL is shipped with pre-packaged Apache SSL certificate to get you started, Zmanda recommends you purchase (or create your own self-signed) SSL certificates and distribute them to all the browsers from which you wish to access the ZMC. The pre-packaged certificates are not secure (as they are shared by all ZRM for MySQL customers). These generic certificates will also generate security warnings on some browser versions.

Zmanda recommends that you either 1) Create self-signed certificates and distribute them to all the client machines that require access to the ZMC, or 2) Distribute certificates from a recognized Certificate Authority. Option 1 (self-signed certificates) is free, and is adequate for most organizations that deploy ZMC servers and the machines that access them **behind the same firewall**.

If using a certificate from a recognized Certificate Authority, your browser will automatically create the secure connection with no errors or warnings.

If using a self-signed certificate, you must then deploy a mechanism to get the relevant browser(s) to accept this new root CA. One method is to generate the certificate using a special format that can be directly imported by common web browsers, and then providing a link on a secure intranet for ZMC users to download (web browsers automatically display the import dialog if the file is in the correct format and sent by the intranet web server using the correct mimetype). PKCS12 (now part of OpenSSL, provides a

mechanism to distribute self-signed private key certificates in a number formats recognized by different browsers.

Another approach is to manually add the new self-signed root CA to the root CA list of the client system, which will automatically provide access to the new CA for all web browsers on the client system. This article covers the procedures for doing this in a Microsoft Windows server environment.

For more details on certificate validation issues, see this article from OpenSSL.

File Locations on ZRM Server

ZRM for MySQL files are installed in following directories:

ZRM Server File(s)	Location
CLI Executables	**/usr/bin**
Man pages	**/usr/share/man**
GUI/Web Components	**/opt/zmanda/zrm**
License key file	**/etc/zmanda/zmanda_license**
Backup Set and other configuration files	**/etc/mysql-zrm, /etc/cron.d, /etc/logrotate.d**
Plugin templates	**/usr/share/mysql-zrm/plugins**
README files and other offline documentation	**/usr/share/doc/MySQL-zrm-enterprise-*** or **/usr/share/doc/mysql-zrm-enterprise**
Libraries	**/usr/lib/mysql-zrm**
Log files (including ssh-copy plugin output)	**/var/log/mysql-zrm/mysql-zrm.log**
Backup images and catalog (default)	**/var/lib/mysql-zrm**
Zmanda Management Console	**/opt/zmanda/zrm, /opt/zmanda/common**

File Locations on MySQL server

MySQL Server File(s)	Location
Binary executables and scripts (Windows)	**C:\Program Files\Zmanda\Zmanda Client for Windows\bin**
Log files	Socket copy plugin output is logged to **/var/log/mysql-zrm/socket-server.log** on the client. There are other log files depending on the type of backup

Caution: Do not directly delete or change any of these files or directories unless directed to do so by the Zmanda Support Team. Changing any of these files directly can result in failed backups and other problems.

Logging in to the Zmanda Management Console

The ZMC is web-based, and can be accessed from any machine on the network. FireFox 3.5 or greater is the recommended browser.

To access ZMC, open any web browser and point it to the ZMC server. For example. if the backup server is *ZMCbackupserver.company.com*, and you have specified port *1234* for web services during the installation process, use the following URL:

https://ZMCbackupserver.company.com:1234

By default, the ZMC web server uses port 443.

If you are running Amanda Enterprise and ZRM Enterprise products on the same server, you will get a product selection page.

Initial User name and Password

After the you have downloaded and installed the ZMC, log in with the default user name "admin" using a password of "admin".

Important note: We strongly recommend that you change the password as soon as possible.

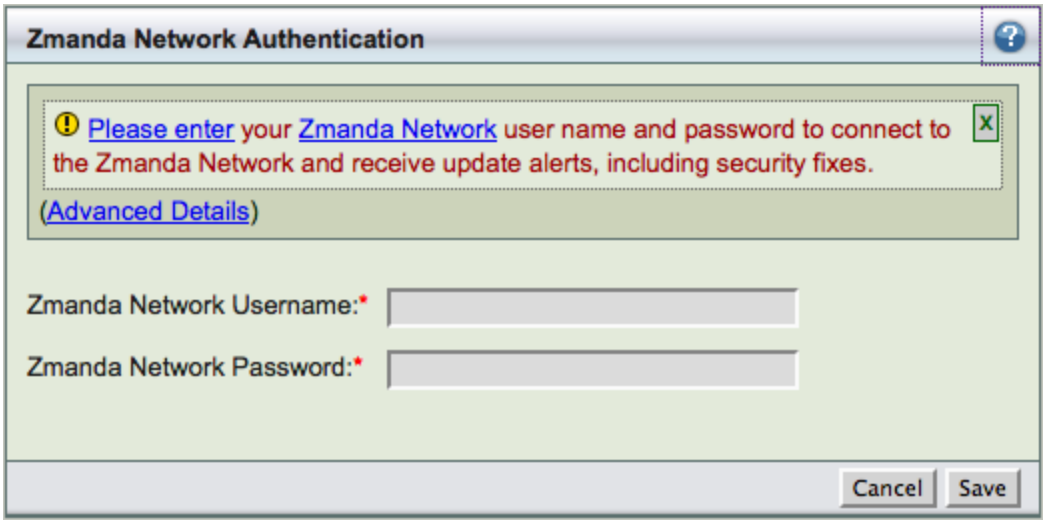
Lost Password

If you do not remember the password, please click **Lost your password?** link. Please enter the ZMC user name in the **Lost Password** section. Please note that password will be mailed to the mail address registered to the ZMC user account. Please note that email service should be configured on the ZRM backup server to receive the lost password email.

If you have difficulty resetting the password, please contact Zmanda Support.

Zmanda Network Authentication

ZMC will ask Zmanda Network User name and password if you login as **admin** user. This authentication allows ZMC to connect to Zmanda Network to authenticate the user as well as obtain alerts including security alerts.


Zmanda Network Authentication
Please enter your Zmanda Network user name and password to connect to the Zmanda Network and receive update alerts, including security fixes.
(Advanced Details)
Zmanda Network Username:*
Zmanda Network Password:*
Cancel
Save

Zmanda Network Authentication is not supported when Web proxy server is in use. If there is a web proxy server or there is no internet connection on the Amanda server, please select **Cancel** button.

Zmanda Network Authentication is performed every time ZMC **admin** user logs in.

About Backup Sets

The backup set is a grouping mechanism that simplifies and optimizes backing up MySQL databases, and tables that are accessible for a MySQL server or is a part of MySQL cluster. It lets an administrator define a set of backup policies (what, how, where and when) to automatically schedule different backup runs.

All ZMC actions (backup, restore, reporting, and monitoring) are performed in the context of backup sets.

A backup set cannot include more than one MySQL server, unless those servers form a cluster. A backup set can include one or more databases. When selecting individual tables as backup sources, you must select a single database, then the tables it contains. A single backup set cannot contain tables from multiple databases.

Multiple backup sets are useful for protecting a large number of systems with different backup requirements, but many organizations with less complex backup requirements can define a single backup set to meet their needs. For example, on a network that includes several databases with high transaction rate along with other databases that change more slowly, you would probably want to create one backup set for the more active databases, and another backup set for the less active ones.

What a Backup Set Contains

A backup set is defined by the following properties:

Name
Besides being unique, the name must consist of alphanumeric characters. Dashes (-) and underline (_) characters are also allowed.

What
Type and Name of MySQL host: Identifies the MySQL server (identified by DNS name or IP) or cluster being backed up, and the database(s), or table(s) within in a database to back up.

Where
The destination directory where the backups will be stored until they are expired by the **retention date** specified by the retention policy.

When
Specifies the backup cycle policies used when automatically generating backup schedules.

How
Specifies the backup method to use and other parameters such as Backup Mode, Replication, Encryption, snapshots, etc.

Multiple backup set configurations

ZRM for MySQL employs multiple levels of default inheritance to simplify the process of administering multiple backup sets:

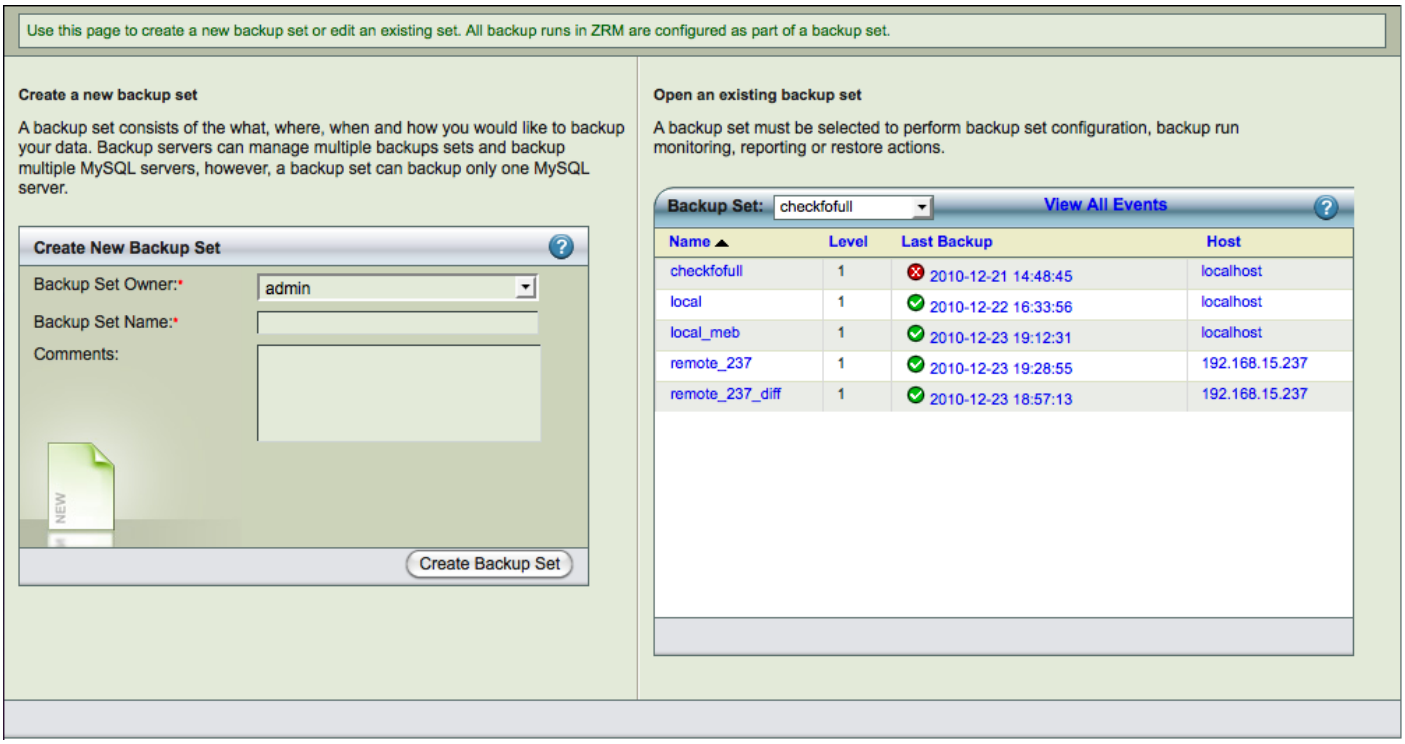
Factory settings
These are the "built-in" assumptions that allow many administrators to use ZRM for MySQL right "out of the box" with few configuration changes.

Site Settings
These allow the admistrator to set global defaults used when creating backup sets. For example, if you have a single MySQL server and are creating multiple backup sets for different the databases it contains, you can set up a site-wide default MySQL server (and related parameters).

Backup set
The backup set itself allows you to override either Factory or Site settings.

Backup Set Starter Page

This is the first page of Zmanda Management Console (see below figure). The left panel can be used to create a new backup set. The right panel is the backup set dash board that shows the list of backup sets, status of the backup set backups and which MySQL server is being backed up.


Use this page to create a new backup set or edit an existing set. All backup runs in ZRM are configured as part of a backup set.
Create a new backup set
A backup set consists of the what, where, when and how you would like to backup your data. Backup servers can manage multiple backups sets and backup multiple MySQL servers, however, a backup set can backup only one MySQL server.
Create New Backup Set
Backup Set Owner:*
admin
Backup Set Name:*
Comments:
Create Backup Set
Open an existing backup set
A backup set must be selected to perform backup set configuration, backup run monitoring, reporting or restore actions.
Backup Set: checkfofull
View All Events
Name
Level
Last Backup
Host
checkfofull 1 2010-12-21 14:48:45 localhost
local 1 2010-12-22 16:33:56 localhost
local_meb 1 2010-12-23 19:12:31 localhost
remote_237 1 2010-12-23 19:28:55 192.168.15.237
remote_237_diff 1 2010-12-23 18:57:13 192.168.15.237

Create New Backup Set

Backup Set Owner

Use the dropdown menu to choose the ZRM for MySQL user who will own the backup set. The only available user will be **Admin** until you configure more users.

Backup Set Name

Specify an unique and descriptive name for the backup set. The name can include any alphanumeric characters, along with periods and hyphens. Spaces are not allowed. Backup Set Name cannot be changed after creation.

Comments

Enter an optional comment that describes the purpose of the backup set. The comment can contain any alphanumeric characters. Special characters such as # and newline are not allowed.

Create Backup Set

Click the **Create Backup Set** button when you are done. The Backup Summary page is then displayed, allowing you to set further options for the backup set.

Backup Set Dashboard

Backup Set Name

The list of backup sets configured on the ZRM server. They can be sorted. You can select a backup set by clicking on the name.

Last Backup Level

The level of backup performed last for the backup set. 0 means full backup. 1 means log incremental or differential or chained differential backup.

Last Backup Date stamp

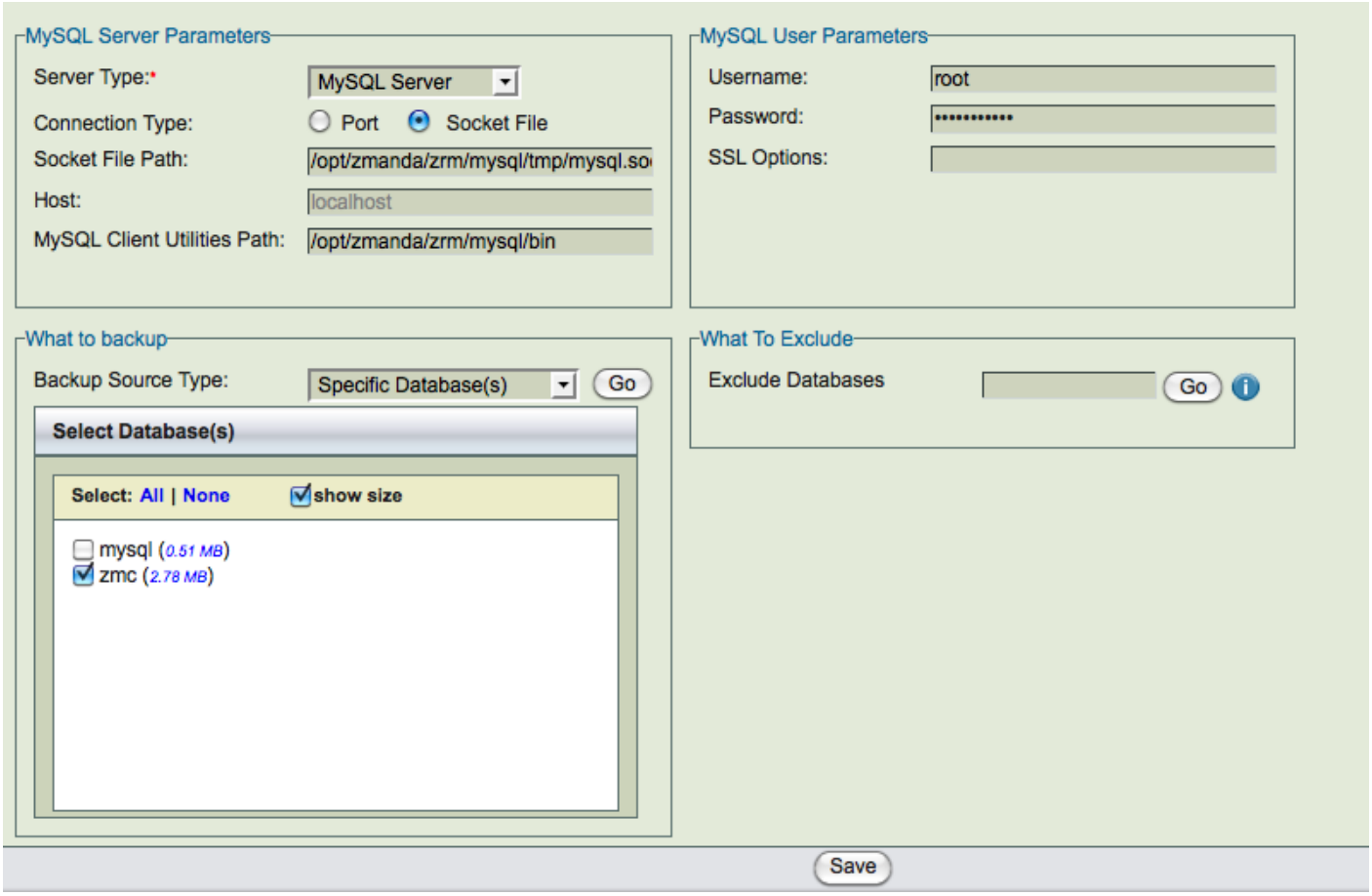
Date and time of last backup performed for the backup set. The green icon indicates the last backup was successful. The Red icon indicates last backup was a failure and the backup set needs attention.

Host

Each backup set is associated with a MySQL server. The host name or IP address of the MySQL server.

Specifying what to Back Up

The **Backup What** page lets you define what databases and tables from a particular MySQL server to back up in the current backup set. To define multiple MySQL servers requires multiple backup sets.



As with many of the Zmanda Console Management forms, the fields change depending on the context; selecting different values can change the options that are displayed.

Note also that backup sets inherit default values from the **Set Site Defaults** page, which in turn will default to factory settings unless you change them. Text fields show inherited defaults on a gray background (see Host parameter **localhost** in the example screen shown above).

Server Parameters

The first group of options (Server Parameters) let you specify the connection details for the MySQL Server to back up. **MySQL Server** is selected as the Server Type by default, and that is what is documented here. If you choose MySQL cluster, **Backup What** displays different options, which are described on the Backing Up a MySQL Cluster page.

Connection Type

Choosing **Port** lets you enter a port number for communication with the MySQL server; choosing **Socket File** lets you enter a path to the socket file.

Host

Enter the host name or IP address of the MySQL server being backed up, or **localhost** if the MySQL server and the ZRM server are on the same machine. Binary logs must be enabled for the MySQL server.

MySQL Client Utilities Path

If you have installed the same versions of MySQL client commands on the ZRM server and the MySQL server (typically in **/usr/bin**), use the factory (Or Site Settings) defaults for all backup sets. If you have installed different versions of MySQL on the ZRM server and the MySQL server (i.e., the backup client), enter the full path to where MySQL binary commands are installed on the backup client. For example: **/opt/lampp/bin**.

MySQL User Parameters

User Name

Enter the username of the MySQL User on the MySQL server. If no user is specified here, ensure that you have the MySQL backup user (with appropriate privileges) specified explicitly as a Site Default, or that the MySQL user specified in the **my.cnf** or **--options file** has the necessary privileges described in System Requirements.

Password

Enter the password for the MySQL User. If you wish to prevent the password from being sent in plaintext over the network, you may wish to configure the MySQL username and password using the MySQL **my.cnf** or **--options file**. The following characters are allowed in passwords: **a-z A-Z 0-9 _ - / . = " ' + *** and space. For the remaining criteria please see this MySQL documentation.

SSL options

This field is valid only for logical full backups. If the connection from ZMC server to the remote server is via SSL, specify the MySQL **ssl** parameters here. If you are backing up databases on the **localhost**, leave this field empty. Please see configuring SSL connection between MySQL server and ZRM server.

What to Backup

Choose an option from the dropdown menu. Note that because the database(s) and table(s) shown on this page are updated dynamically based on information retrieved from the MySQL server, these may not match what has been set on the **Site Settings** page. The **Go** buttons allow you to refresh the connection to the MySQL server(s) to determine what databases and tables are *currently* available for backup. Available options are:

All Database(s)

Choosing this option includes all databases and all tables for backup.

Specific Database(s)
Choosing this option displays a list of databases available for backup on the MySQL server. Check all the databases that you want to add to this backup set. Size of the databases are also displayed if **show size** checkbox is selected.This will help determine the size needed for full uncompressed backup images.

Specific Table(s)
Choosing this option displays a dropdown menu that lets you select a database from which to select tables to backup, followed by the list of tables in the selected database; check all the tables that you want to add to this backup set. Size of the tables are also displayed if show size checkbox is selected. This will help determine the size needed for full uncompressed backup images.

```
Note: Database names and table names cannot contain # character
```

What to Exclude

Exclusions can be used to backup all databases except few or all tables in a database except few that match the specified pattern. This is useful when a MySQL server has mixture of production and test databases and test databases need not be backed up. It can also be used to split large environment (lot of databases or lot of tables) into multiple backup sets.

The pattern applies to database names if all databases or specific databases are in the backup set. The pattern is matched with table names if a specific tables are in the backup set.

The wildcards that are supported are:

*****
matches one or more character. For example: zmanda*db pattern will exclude *zmanda_bugs_db*, *zmanda_wiki_db* and will not match *zmanda_network* names.

?
match only one character. For example: zmanda_db? will match *zmanda_test1* and they will be excluded from the backup set.

[<char><char><char>...]
match any character. For example: test[123] will match *test1*, *test2* and *test3*.

|
match one of the patterns. For example: cat*|dog* will match all names that begin with *cat* or *dog*.

Backing up MySQL Application-specific files

MySQL applications such as Sugar and MediaWiki consist of MySQL databases and associated configuration files. Backing up the databases alone will not completely protect these application servers from mishaps. For this reason, Zmanda allows you to add configuration files to backup, however the ZMC GUI does not yet

support this. To add such application-specific files to the backup, edit the MySQL configuration file, **mysql-zrm.conf** for the given backup set, and use the **--config-file-list** option to mysql-zrm-backup(1) command to specify the files you wish to backup.

All application files that are backed up are compressed or encrypted depending on the backup set configuration. Backup of configuration files specified by *config-file-list* in the *mysql-zrm.conf* are backed up only during full backups. Incremental backup of these files are not performed. The backup is performed as *mysql* user. So, *mysql* user should have permissions to read these files.

This functionality is available only on Linux and Solaris platforms.

To use this functionality:

- Define *config-file-list* parameter in the *mysql-zrm.conf* for the backup set. Edit */etc/mysql-zrm/<backup set name>/mysql-zrm.conf*. for example:

```
config-file-list="/etc/mysql-zrm/<backup set name>/application-configuration
```

  The configuration file can be in any directory in the ZRM server.

- (optional) It might be useful to generate the application configuration file dynamically before the backup run. This will allow all application files to be part of the backup set. The generation of configuration file can be done using pre-backup-plugin. Pre-backup plugin can be configured in ZMC Backup|How page.

The pre-backup plugin can be use *find* command to generate list of files in the application directory. You can specify list of directories or files to exclude to *find* command. Following script can be used to backup ZMC dependency configuration files. This script can be used as a pre-backup plugin.

```
#!/bin/sh
#
# List of directories to backup
#
dirs="/opt/zmanda/zrm /opt/zmanda/common";
#
# List of sub-directories to be excluded from the backup
#
exclude_dirs="/opt/zmanda/zrm/tmp
              /opt/zmanda/zrm/php/tmp
              /opt/zmanda/zrm/mysql/tmp";

#
# List of files that should be excluded
```

```
#
exclude_filename_wildcards="*.tmp *svn*";

#
# configuration_file: List of files to be backed up. Do not change this value wi
# fixing mysql-zrm.conf
#
configuration_file="/etc/mysql-zrm/zmc-backup/zmc-configuration-files";
excl_fname_pattern="";
for exclude in $exclude_filename_wildcards;
do
       excl_fname_pattern="$excl_fname_pattern -o -name '$exclude'";
done

# strip the first -o
excl_fname_pattern=`echo $excl_fname_pattern | sed -e 's/-o//1'`;

excl_dir_pattern="";
for exclude in $exclude_dirs;
do
       excl_dir_pattern="$excl_dir_pattern -o -type d -path $exclude -prune";
done

rm -f $configuration_file

for dir in $dirs;
do
       find $dir -type f -not '(' $excl_fname_pattern $excl_dir_pattern ')' -pri
done

exit 0;
```

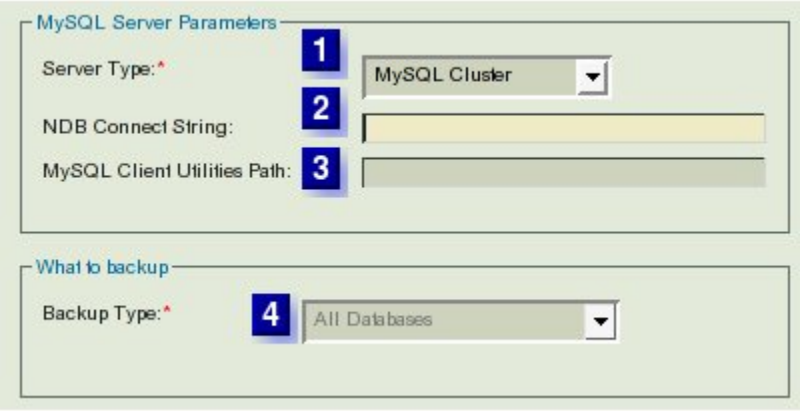
MySQL cluster (NDB storage engine) backup/recovery

- Please note that MySQL cluster is backed up using NDB management tools (*ndb_mgm* and *ndb_restore*) tools. NDB management tools are required on the ZRM server.
- Full backup of each NDB data node is performed on each data node in the backup directory (configured in **Backup|Where** page). The backup directory must be present in each NDB data node and cannot be the same directory across nodes. The backup is then copied to ZRM server using copy plugin. The ssh or socket copy plugin must be configured in each NDB data node (See **Backup|How** page).

- Incremental backups of MySQL cluster cannot be performed on NDB data nodes. Since there can be multiple SQL nodes in a NDB cluster, the incremental backups cannot be performed correctly.

Backup What page

By default, ZRM for MySQL shows the options for a MySQL server on the **Backup What** page. When you choose **Cluster** as the server type, options appropriate for that choice are displayed:



NDB Connect String

This identifies the cluster to connect with for making the backup. To back up multiple clusters, multiple backup sets are required, each set with the appropriate NDB connect string. The connect string is not validated when you save the settings, so be careful to enter a valid connect string if you want to avoid subsequent backup failures.

Consult the [MySQL Cluster documentation] for further details.

MySQL Client Utilities Path

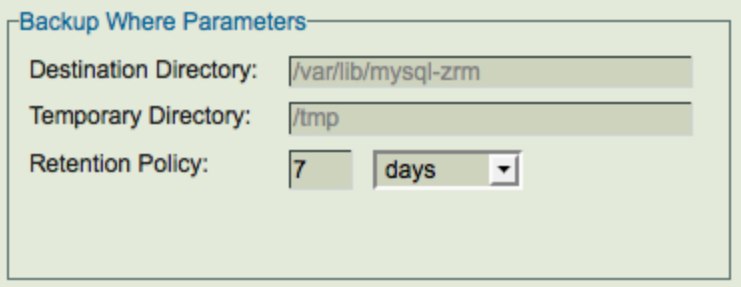
The location of MySQL binary commands used to manage the cluster server.

What to Backup

When performing backups on clusters, all databases are backed up; the **What to backup** option is grayed out and disabled.

Backup Where

This page specifies where the backup images for the backup set will be stored, and how long to retain them.



Destination Directory

Regardless of where the remote MySQL server is hosted, backups are stored under the backup directory of the local machine where ZMC runs. The default **/var/lib/mysql-zrm**. If you specify another directory, you must create it yourself on both the ZRM server and the MySQL server, and ensure that its permission settings allow read/write access to the mysql backup user. The directory used on the MySQL server and ZRM server cannot be the same i.e cannot be the same CIFS or NFS share.

The recommended practice is to mount a filesystem at **/var/lib/mysql-zrm.** Use local, NFS, or CIFS mounted storage for storing MySQL backup data. The MySQL backup data can be migrated to other storage devices using Network-based backup and recovery utilities such as Amanda. If you are using NFS or CIFS storage as Destination directory, the root user and mysql user should have privileges to read and write to this directory.

You must allocate sufficient disk space to store the MySQL databases. If sufficient disk space is not available, the backup run will fail. The destination directory should have at least 150% of uncompressed backup space available for a successful backup run. This additional space is required even when backups are configured for compression, since the additional space is used for compression during the backup run. The additional space is freed after the backup run.

Example: Backup set "daily" requires 150GB of disk space for uncompressed backups and backup image upon compression are 100GB. The backup directory filesystem should have at least 225GB (1.5 * 150GB) for space. After backup run is completed, only 100GB of disk space will be used.

The destination directory must not be under autofs controlled mount point because snapshot based backups mount the file system or storage snapshot under the Destination directory.

Temporary Directory

ZMC creates temporary files and directories during backup and restore. How much space this requires in practice depends on the backup method.

Logical backup/Snapshot backup - Few kilobytes are required during backup and size of full backup image is required during restoration.

Raw backup (using mysqlhotcopy) and MySQL Enterprise Backup tool - Size of full backup image plus 10% is required during backup as well as restoration

Xtrabackup tool - Size of full backup image during backup. During restoration of full backups, backup image size is required. If restoration involves full backup, differential/chained differential backups, size of temporary directory must be sum of all backup images.

The default is to use the OS-specified temporary directory ($TMPDIR on Linux). It is recommended that the default value is changed so that sufficient temporary space if available for backup and restoration. In any case, the directory must exist on both the client and server, and have permissions set to allow read/write access to the MySQL backup user. If $TMPDIR is defined differently on the ZRM server and the MySQL server, you must explicitly set the path here and make sure that directory exists on both machines. In case of Solaris /tmp is memory based file system and filling it up will cause processes to run out of memory. So, default value must be changed.

Retention Policy

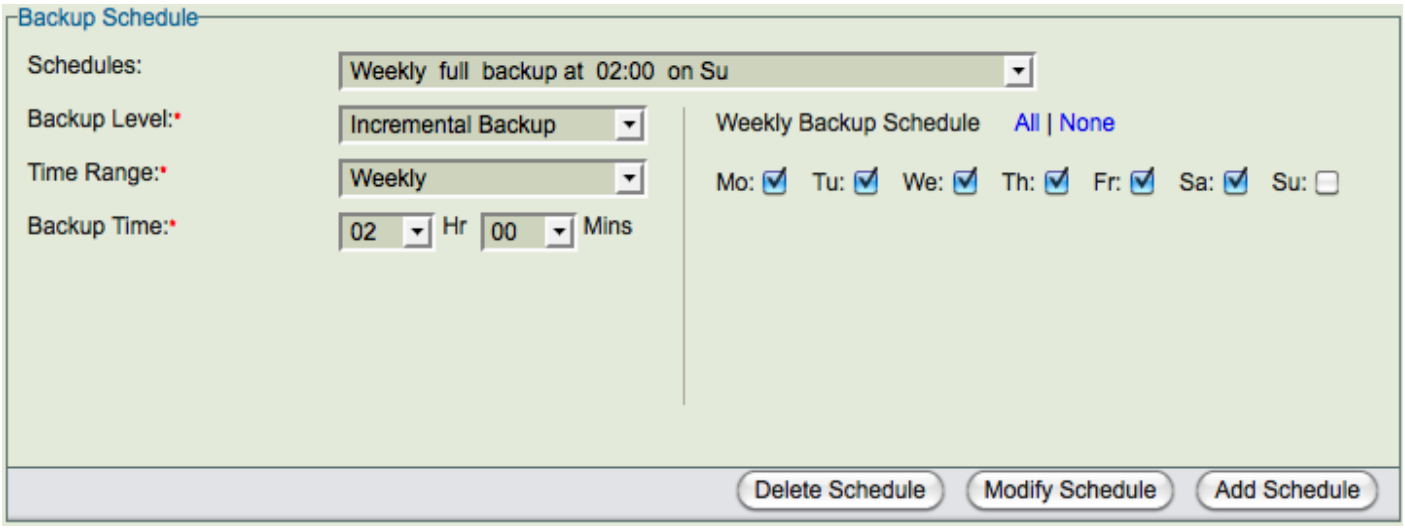
A retention policy sets the limit of the period for which the backup set will be retained. Images older than the retention policy specified are automatically purged. The retention policy for a backup image is stored in the backup index at the time of backup. So, changing the retention policy will not change the retention policy of backup runs that have been completed. If the retention policy is not specified, the backup images are retained forever.

Backup When

This ZMC page allows users to schedule backup runs for the backup set. Users can add a schedule a new backup run (**Add Schedule**), modify an existing scheduled run (**Modify Schedule**) and delete an existing scheduled run (**Delete Schedule**).

ZRM uses operating system **crontab** tool to implement this functionality.

The figure below shows a backup schedule that is being created. There is a weekly full backup scheduled at 2am (local time on the ZRM server) on Sundays. A Log incremental backup is being added on other days at 2am.

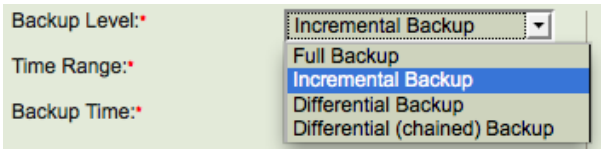


Schedules

List of schedules that have been configured for the backup set. If there are no scheduled backup runs, **Add New Schedule** is displayed. ZRM does not check to see if the scheduled backup runs overlap. Overlapping backup runs can be scheduled for certain backup methods.

Backup Level

There are various backup levels as shown below.



Full backup means all databases/tables in the backup set (specified in **Backup What** page) are backed up. It also backs up all binary logs from the MySQL server. The method used for Full backup is determined by **Backup How** page parameters.

Log incremental Backup is available for all backup methods. The incremental backup will contain all binary logs since the last full backup or last incremental backup. This backup type requires Binary logs to be enabled on the MySQL server. The incremental backup in earlier ZRM versions is log incremental backup.

Differential Backup is available only for MySQL Enterprise Backup. This backup contains all block level changes since last full backup. Using this backup incremental type when compared to Chained Differential incremental type, makes the restoration easier but the backup size becomes larger for every successive Differential backup.

Chained Differential Backup is available only for MySQL Enterprise Backup. This backup contains all block level changes since last differential backup or full backup.

Time Range

The values can be **Daily**, **Weekly**, **Monthly**. Weekly backup allows you select the list of specific days and Monthly backup allows you select the list of specific dates in a month. Hourly backups can be scheduled by selecting Daily backup and specifying various backup times.

Backup Time

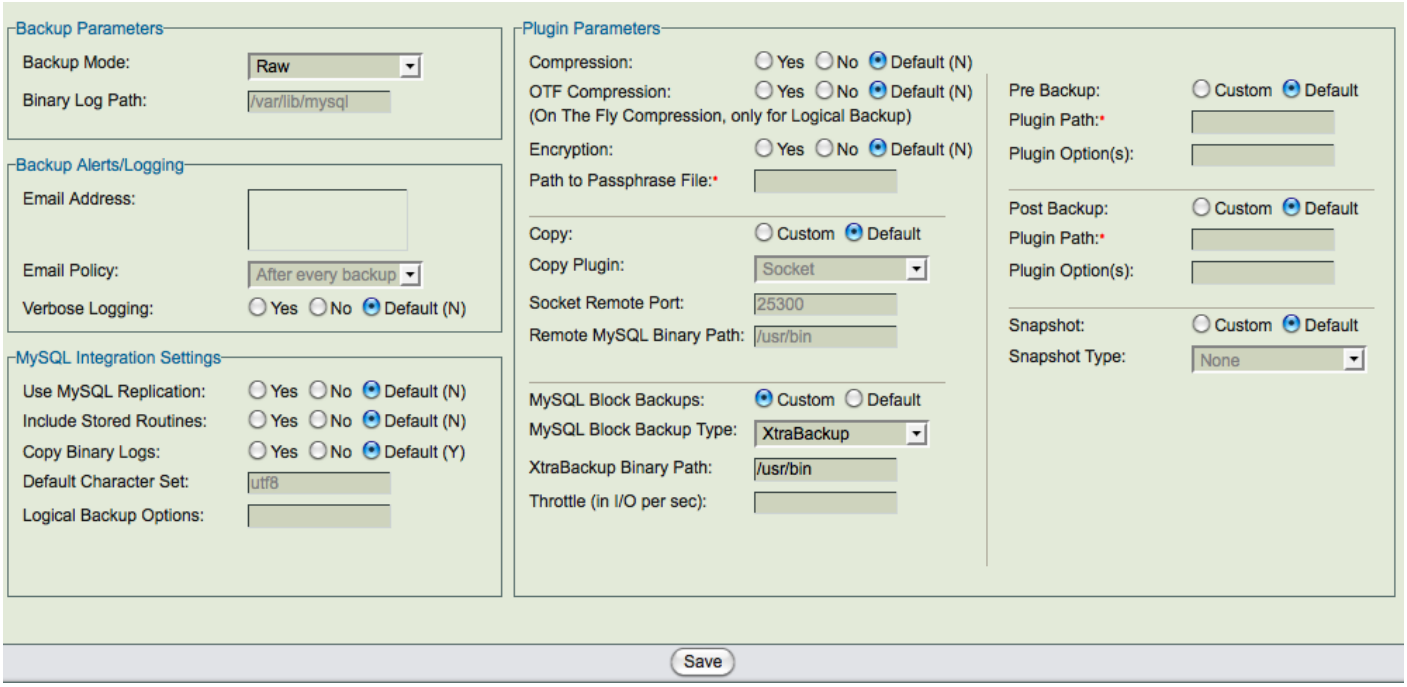
Specific time of the day (24 hour clock) when the backup run starts. The local time zone of the ZRM server is used.

Backup How

The **Backup** how page lets you specify the mode of backup (**raw** or **logical**), along with other options that let you optimize the type of backup given the performance required at your site.

The page is divided into panels that group settings according to function (**Backup Parameters**, **Plugins**, etc.). Most of the settings have default values, inherited from the **Site Settings** page or from factory defaults. Default strings are shown in the text field with a gray background. Radio buttons options (**Yes**, **No**, **Default**) show the current default value in parentheses to the right of the Default radio button (**Y** or **N**).

An example Backup How page is shown below (default options). When **Raw Backup Mode** is selected, Snapshot panel will also appear



Backup Parameters

Backup Mode

There are two possible values:

Raw

A raw backup makes a copy of the binary disk image of databases stored on non-transactional storage engines by using mysqlhotcopy. Because the database must be quiescent while any type of backup is in progress, raw backups often provide shorter backup windows, especially if a snapshot plugin is used. Although raw backups can be restored more quickly than logical backups, they can only be restored to same version of MySQL server on the same platform architecture. If any of the databases or tables are stored on a transactional storage engine (such as InnoDB), a logical **mysqldump** backup is taken instead unless snapshot backup method is configured.

When you select **Raw Backup Mode**, the snapshot options are displayed in the page.

If the snapshot backup fails, ZRM automatically defaults to Raw backup using **mysqlhotcopy** command if the backup set contains only non-transacational storage engines such as MyISAM. ZRM tries to ensure a successful backup method is performed. To disable this feature, set

fallback-for-full-backup=0

in the **/etc/mysql-zrm/<backup set name>/mysql-zrm.conf*** configuration file for the backup set in the ZRM server.

Logical

Choosing this option forces a **mysqldump** backup that copies MySQL binary logs regardless of the storage engine. MySQL binary logs track and save all database server transactions as a list of SQL statements. To implement a logical backup strategy, binary logging must be enabled on the MySQL server, and a path to the log files must be supplied (the default is **/var/lib/mysql**) in the **Binary Log Path** field.

Logical backup method works with all MySQL storage engines except the MySQL cluster NDB storage engine. Logical backups can also be restored to any platform architecture or database that supports

SQL. For example: Backups of MySQL database running on Redhat Enterprise server powerpc platform can be restored to Ubuntu server running on x86 platform.

Because logical backups require a read lock on the database(s) or tables being backed up, they can have a greater impact on the applications using the MySQL database. Logical backups also result in increased restore times, as restoring the data is accomplished by re-playing the transactions against the target database instead of just copying files.

If the snapshot backup fails, ZRM automatically defaults to Logical backup using **mysqldump** command if the backup set contains transacational storage engines such as InnoDB. ZRM tries to ensure a successful backup method is performed. To disable this feature, set

fallback-for-full-backup=0

in the **/etc/mysql-zrm/<backup set name>/mysql-zrm.conf*** configuration file for the backup set in the ZRM server.

Binary Log Path

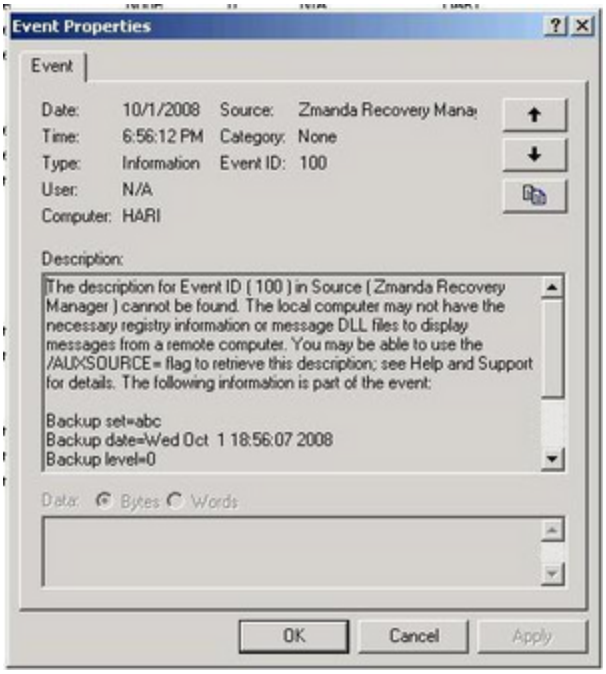
Enter the full path where the MySQL binary logs are stored once binary logging has been enabled. If nothing is entered here, the site default path is used.

Backup Alerts/Logging

Email Address (or Update Windows Event Log on Windows servers)

On Linux/Solaris servers, enter the email address of the ZMC backup administrator. All backup run notifications and summary reports will be automatically sent to the email address based on the email policy. A mail user agent (MUA) such as **Mailx** must be manually configured on the MySQL server before any email can be sent. To email reports and notifications to more than one user, set up a group email alias on your mail server and enter group email alias here. Alternatively, you can enter multiple email addresses separated by space. For example: **admin1@company.com admin2@company.com**

On Windows ZRM servers, select whether to use the Windows Event Log to save and display notifications and summary reports. Note that even when logging successful ZRM operations, the Windows Event Log will show the following error.


Event Properties
Event
Date: 10/1/2008 Source: Zmanda Recovery Mana
Time: 6:56:12 PM Category: None
Type: Information Event ID: 100
User: N/A
Computer: HARI
Description:
The description for Event ID (100) in Source (Zmanda Recovery Manager) cannot be found. The local computer may not have the necessary registry information or message DLL files to display messages from a remote computer. You may be able to use the /AUXSOURCE= flag to retrieve this description; see Help and Support for details. The following information is part of the event:
Backup set=abc
Backup date=Wed Oct 1 18:56:07 2008
Backup level=0
Data: Bytes Words
OK
Cancel
Apply

The "Event description not found" messages can be safely ignored.

Email Policy

This policy determines when email should be sent to the configured **Email Address**. Policy can be **After every backup** (the default value), **Never** (no notification is performed) or **Backup with error** (when there are failures in the backup run).

Verbose

Verbose logging should turned **on** or **off**. Zmanda Support Team will ask you turn on verbose logging for backup sets to troubleshoot problems in a backup set.

MySQL Integration Parameters

These options allow you to optimize backup performance by selecting from the various options described below.

MySQL Replication

Specifies whether ZRM for MySQL will use the MySQL Replication facility that has been enabled on the MySQL server that is to be backed up. Backing up a replication slave reduces impact on the production database near zero while backups are in progress. When this option is enabled and the mMySQL server is a replication slave, ZRM for MySQL will back up the **master.info**, **relay-log.info** and any **SQL_LOAD-*** files if they exist. For further details on the benefits of using a replication slave for backup, see this MySQL documentation. The **master.info** and **relay-log.info** files are described here.

Dump Store Routine

To improve the performance of database functions and procedures, MySQL versions 5.0 and higher allow users to compile and store them as reusable routines (Stored routines). This option specifies whether stored routines should be include during logical backups. The default value is **No**. If your version of MySQL supports this feature, it should probably be set to **Yes**. Setting the value to **Default** implies Site specific value is used for this parameter.

Force InnoDB Backup

Choose whether to force backup of InnoDB logs and shared data regardless of whether any tables in the backup set are stored by InnoDB. If set to **No**, ZRM for MySQL performs a check, and does not back up InnoDB logs and shared data unless the backup set has tables stored by InnoDB. In most cases, the default of **Yes** provides the best performance, because the check takes longer than just backing up the InnoDB tables.

Default Character Set

Specify the default character set that is used in the MySQL database. The default value is **utf8**; if the database uses a different character set, reset accordingly.

Logical Backup Options

Parameters to the **mysqldump** MySQL command. Logical backup uses **mysqldump** command. You can customize the options using this field. For example: "*--max_allowed_packet=1G*" can be specified as value. These parameters are used in addition to parameters passed by ZRM.

Plugin Parameters

Plugins allow you to specify various operations before before, during, and after the backup. Plugins are available for compression, encryption, and what copy mechanism to use. You can also integrate your own pre- and post-backup plugin operations.

Compression

Backup compression is performed on the ZRM server. Allows you to compress the image using **gzip** on the ZRM server. If the storage engine already compresses data (such as ARCHIVE storage engine), there is no value in setting this to **Yes.**

OTF (On the Fly) Compression

(**Applies only to logical backups**) Specify whether backup image files stored on disk should be compressed as the backup progresses. The default is to start compression only after all backup files have been saved to disk. Turning this option on can save disk space, but results in slower backups.

Encryption

Backup encryption is performed on the ZRM server. Default encryption is performed using GPG. ZRM server should have GPG (GNU Privacy Guard) packages installed. These packages are part of Linux and Solaris distribution. The default encryption algorithm used is AES 256. If different key length or cryptographic algorithm is required, please modify gpg parameters in the script */usr/share/mysql-zrm/plugins/encrypt-plugin.pl* on the ZRM server. Please contact Zmanda support team for help.

Path to passphrase file

The full path to the encryption passphrase file. The encryption file is usually stored as */etc/mysql-zrm/<backup set name>/.passphrase* on the ZRM server. The passphrase should be owned by **mysql** user and should have 400 permission. It is very important to protect the contents of the passphrase file. Passphrase file contains the sequence of words or a string that is used for encryption. It is important to choose a good passphrase. The longer and more random the passphrase, the more difficult to crack the encryption. The passphrase file should contain the passphrase in the first line (other lines in the file are not read). A method to generate passphrase is to use the following command:

```
$ gpg --gen-random 1 16 | gpg --enarmor | sed -n 5p > /etc/mysql-zrm/<backup set
```

Backup Encryption Caution: It is very important to store the encrypt-plugin.pl a
for restoration. Without these files, it is impossible to restore the backup ima

Copy

The copy plugin is used to transfer backup files to and from a remote MySQL server to the ZRM server. Use this option in the following circumstances:

- backing up any Windows-based MySQL server; use the **Windows** copy plugin.
- remote incremental backup is required
- copying replication-related files from a remote machine
- executing **mysqlhotcopy** (MySQL command) to copy the data from the remote MySQL server
- executing **mysqlhotcopy** to restore the data to a remote MySQL server.
- backing up remote machines via snapshot backups.

For non-Windows remote MySQL servers, choose **SSH** or **Socket**, and then enter the appropriate parameters for the option selected. Available copy plugins are described below:

SSH

Secure Shell-based communication. Use when a higher level of security is required (such as when the client and server communicate across a firewall). You must then enter the SSH username on the remote MySQL server, and the path where MySQL commands are installed on the remote machine. For backup, it is necessary to set up password-less *ssh* configuration is required (i.e, **mysql** user on ZRM server should be able to *ssh* to MySQL server without being prompted for password). During restore, users will prompted for ssh password if needed. The user id and group id of **mysql** user must be same on the ZRM and MySQL server.

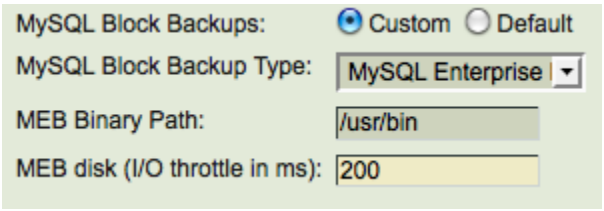
Socket

Socket-based communication. This is the default copy plugin. This requires that the ZRM client components are installed on the remote machine, as described in the Installation Instructions. You must then enter a port for the remote socket (the factory default is **25300**. The path where MySQL commands are installed on the remote machine must be provided. The user id and group id of **mysql** user must be same on the ZRM and MySQL server.

Windows

The Windows copy plugin is required for backing up any Windows-based MySQL server. It requires the ZRM for MySQL Windows client components described in the Installation Instructions. You must then enter the communications ports to use during backup (default is 10080) and restore (default is 10081) operations

MySQL Enterprise Backup/InnoDB Hot Backup - MySQL Block backups

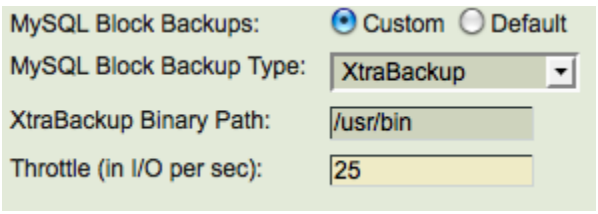


Select MySQL Enterprise Backup as the MySQL Block Backup Type to use Oracle's InnoDB Hot Backup or MySQL Enterprise Backup to perform the backup. Oracle's InnoDB Hot Backup must be purchased from www.innodb.com or use MySQL Enterprise Server Standard edition or higher. This option provides integration between ZRM and the Oracle product. This allows backup to proceed without setting any locks or impacting database operation. You must then supply the path to the InnoDB Hot Backup tools: the ibbackup binary and the mysqlbackup tool, which must be installed in the same path on the MySQL server being backed up. The default path is /usr/bin. The *mysql* user should have permissions to execute the mysqlbackup tool. MEB disk throttle can be specified to throttle disk I/O during backup. It is specified in milliseconds, the backup sleeps for that time between disk I/O. Full, differential and chained differential backups are performed using this tool.

MySQL Enterprise Backup requires the innodb_data_home_dir to be specified explicitly in the my.cnf for the MySQL server being backed up. The directory cannot be specified as part of innodb_data_file_path. For example:

```
innodb_data_home_dir = /innodb/innodbdata/
innodb_data_file_path=ibdata1:50M;ibda2:50M;ibda3:40M;ibd4:20M:autoextend
```

Xtrabackup - MySQL block backups



Select XtraBackup as the MySQL Block Backup Type to Xtrabackup tool to perform the backup. Xtrabackup tool can be downloaded from Percona or SkySQL. It is open source and can be downloaded to MySQL server. This option provides integration between ZRM and Xtrabackup. This allows backup to proceed without setting any locks or impacting database operation. You must then supply the path to the Xtrabackup tools: the xtrabackup binary and the innobackupex tool, which must be installed in the same path on the MySQL server being backed up. The default path is /usr/bin. The *mysql* user should have permissions to execute the innobackupex tool. Xtrabackup backups can be throttled by specifying the number of disk I/Os performed by the backup tool in a second. Full, differential and chained differential backups are performed using this tool. This tool also users to restore a table from a database backup. Restores of table can be performed only to Percona MySQL servers.

Pre-backup

Choose **Custom** to activate the feature (unless you have set a site setting; the factory setting is to use no pre-backup plugin), then specify the full path and any command-line options to the script or utility you want executed before the backup run starts. This feature can be used to check and prepare the MySQL server environment for backup. For example, you could use it to notify all MySQL database users that a backup is about to begin.

ZRM for MySQL does not check if the path is a valid path, nor if the plugin is present at the given location. The recommended location for plugins is **/usr/share/mysql-zrm/plugins** directory. A template file (**pre-backup.pl**) is installed in this location; use the template when writing your own pre-backup plugin. All pre-backup plugin scripts must accept the following command-line parameters (which are passed to the plugin using the **Plugin Option(s)** field:

--all-databases
Use when all databases in the MySQL server are being backed up.

--database *database1 database2...*
Use when selected databases are being backed up.

--database *database* --tables *table1 table2 ...*
Use to specify the list of tables in the database that are being backed up.

The script should be written to return a non-zero value on error; pre-backup plugin failures should cancel the backup and generate a failure message for reports and logs.

Post-backup

Choose Custom to activate the feature (unless you have set a site setting; the factory setting is to use no post-backup plugin), then specify the full path and any command-line options to the script or utility you want executed after the backup run is completed. A template file (post-backup.pl) is installed in this location; use the template when writing your own pre-backup plugin. In addition the parameters for the pre-backup plugin described above, the post-backup plugin should be written to accept the following additional options:

--backup-directory *path*

The post-backup plugin is executed twice: before the checksum is performed, and after the checksum is complete. See the man page for mysql-zrm-backup(1) for more details on how the post-backup plugin operates and the flags it returns.

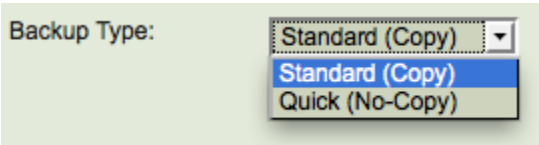
Note that although failures of the post-backup plugin are logged, they are not included in backup reports if the backup itself succeeded.

Snapshot

Select a snapshot plugin if desired. Snapshot plugins are described in more detail in the Snapshot Plugins section of this manual.

Backup Type

This field appears only when Snapshot plugins are used. The values are shown below:



Standard (Copy)

Full and Incremental backups are copied to the ZRM server to the destination location specified in the **Backup Where** page.

Quick (No-Copy)

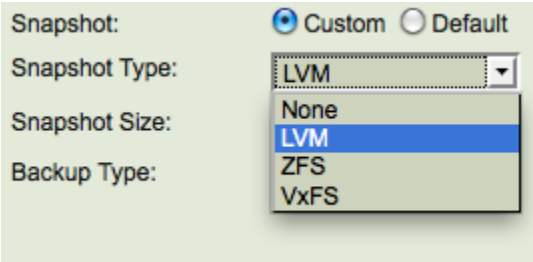
Full backups are maintained as snapshots. Full backup are not copied to the ZRM server. This backup is faster but full backup will not available if the storage system fails. So, it is important to convert quick backups to regular backups. Incremental backups are copied over to ZRM server.

Snapshot Plugins

ZRM for MySQL can be licensed to use various 3rd party snapshot and storage checkpoint mechanisms to acquire a quiescent, consistent copy of the MySQL database, while minimizing application downtime. Unlike other backup methods, snapshots and storage checkpoints scale well; they do not increase the backup window as databases grow.

Because snapshot and storage checkpoint mechanisms are faster than backups to other media, this reduces the time that database tables must be locked. These technologies create a consistent copy of the MySQL database with little impact on MySQL applications, and thus scale well as databases grow. If the MySQL databases or tables use only transactional storage engines such as InnoDB, the time the application is locked is further reduced. While taking snapshots of databases or tables that use non-transactional storage engines such as MyISAM, ZRM for MySQL flushes the database pages to the disk and obtains a read lock on the database(s) / table(s). The read lock is held only for a moment. File system I/O is stopped before taking a snapshot when the database resides on the file systems that support freeze/thaw operations such as XFS, VxFS (Veritas file systems).

Snapshot plugins are configured in the Backup How page. Only snapshots that are licensed appear in the drop down box.



The snapshots are named "zrm<unique string_<yyyymmddhhmmss>". This will allow users to identify when the snapshots were taken.

Supported Mechanisms

ZRM for MySQL supports a number of different snapshot mechanisms provided by OS, filesystem, and storage appliance vendors (follow the links for details on requirements and configuration):

- **VxFS:** The Veritas File System. ZRM for MySQL supports all versions of VxFS (including HP's JFS and OJFS). Depending on licenses purchased ZRM for MySQL can use either a filesystem snapshot or a storage checkpoint for backup.
- **NetApp:** If the database resides on a NetApp storage system (connected using NFS), ZRM for MySQL can use the NetApp SnapManager software to back up a snapshot of the MySQL database.

- **Logical Volume Manager:** On Linux platforms, ZRM for MySQL can use Logical Volume Manager snapshots.
- **ZFS**: ZRM for MySQL can take advantage of ZFS snapshot capability when backing up a MySQL server installed on a Solaris ZFS file system.
- **EMC Snapview**: ZRM for MySQL can take advantage of EMC Snapview capability when backing up a MySQL server installed on an EMC CLARiiON networked storage system.
- **Bluearc Titan**: ZRM for MySQL can take advantage of Bluearc Titan storage snapshot capability when backing up a MySQL server installed on Bluearc Titan storage using NFS.
- **Amazon EBS** : Amazon Elastic Block Storage snapshots to backup MySQL databases running on Amazon EC2.
- **Microsoft Volume Shadow Copy Service (VSS):** On Windows platforms, ZRM for MySQL always uses the Microsoft Volume Shadow Copy Service to perform raw backups; no license is required. There are no special requirements beyond installing the Windows client, which is described in the Windows client section of the installation instructions.

Quick (No-copy) option

When the Quick (No-copy) snapshot option is enabled, ZRM for MySQL uses the snapshot itself as the backup rather than transferring the data into a standard backup archive on the ZRM server.

Quick snapshot backups are appropriate for large databases and for databases that have high transaction rates. In addition to eliminating data transfer bottlenecks during backup, quick snapshot backups also provide much faster database restoration than other backup methods.

Because quick snapshot backups do not copy the data off of the MySQL server, they does not protect data against server media failure. For this reason, quick snapshot backups can be converted at any time into standard backups stored on the ZRM server by using the **Convert Backup** option on the **Reports** menu tab, described in Converting Quick Snapshots to Standard Backups.

Setting Retention Policies for the Quick Snapshot Option

Because using the quick snapshot option for backups is so fast and convenient, administrators tend to schedule many of them. This is fine, so long as you set retention policies to prevent collecting too many snapshots on the MySQL server. Depending on the snapshot technology implemented, retaining an excessive number of snapshot backups can:

- cause the MySQL server to perform more slowly
- cause the MySQL server to run into disk space limitations
- exceed the number of snapshots per server that are allowed by the given technology.

Refer to the documentation provided by the snapshot technology vendor when scheduling and setting retention policies for snapshot backups.

Linux Logical Volume Manager Snapshots

Logical Volume Management (LVM) is a way of virtually partitioning a hard disk space such that it can be flexibly allocated to various applications. It is being utilized by an increasing number of MySQL installations.

ZRM for MySQL has an optional mechanism to help backup such installations using snapshots (a feature license is required). It can create temporary snapshots of the logical volumes and use the snapshot volume to do backups. The advantage of using snapshots is that you need to lock the database tables only for the time taken to create a snapshot. The snapshots are removed when the backups are completed. Snapshots help to create a consistent copy of the MySQL database as the consistency is ensured before the snapshot is taken.

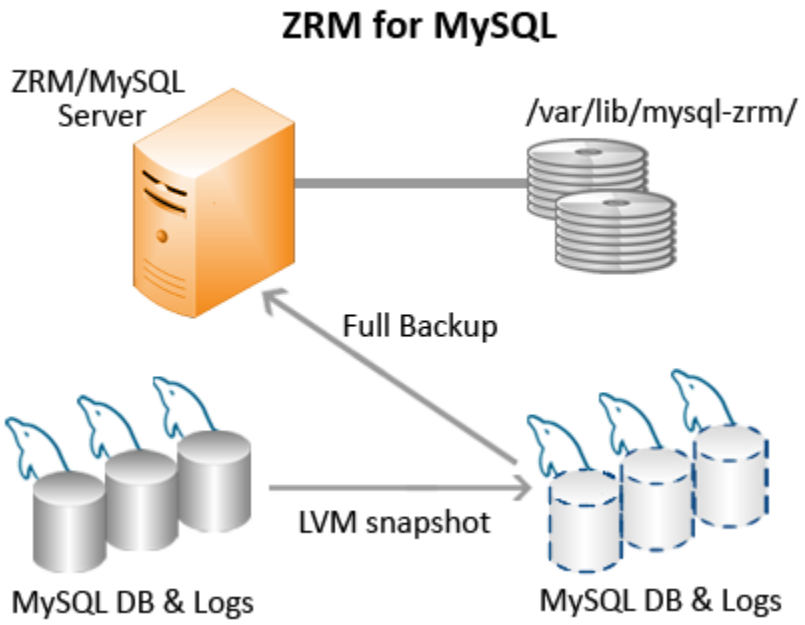
On file systems such as XFS, VxFS (Veritas file systems) that support freeze/thaw operations, file system activity is stopped before taking a snapshot.

MySQL Configuration

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see the next section). On Solaris platforms, this means the **SMCsudo** package must be installed.
- If you are backing up a remote MySQL server, the destination directory specified on the **Backup Where** page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

The mysql data must reside on logical volumes. The following are some of the possible configurations

- All MySQL data is on a single logical volume
- Specific database directories are on different logical volumes
- For databases containing InnoDB based tables, the lvm snapshot can only be used if the database directory, the InnoDB data files, and the InnoDB logs are on logical volumes. The InnoDB shared data files can be on separate logical volumes. The InnoDB logs can be on separate logical volume



Pre-conditions for Using LVM Snapshots

The MySQL backup user must be granted **sudo** privileges to execute **lvm** commands on the MySQL server. Add a line similar to the following example to **/etc/sudoers** on the MySQL server:

```
mysql <FQDN of MySQL Server>=NOPASSWD:/bin/mount,NOPASSWD:/bin/umount,NOPASSWD:/
```

Where *MySQLserver.mycompany.com* is the fully-qualified domain name for the MySQL server. Note that if **lvm** commands are installed in other locations, the above example would not work without editing it to reflect the different paths.

Additional free extents in the logical volume are needed for creating snapshots. You can check extents using the **vgdisplay** command.

The free extents required are specified in mysql-zrm.conf

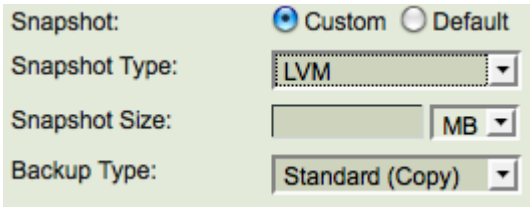
LVM stores the snapshot blocks corresponding to the blocks that are modified in the original logical volume in the snapshot volume. If the database is highly active during the backup, many blocks will be modified and the snapshot volume may run out of space.

Specifying a sufficient amount of space for creating the snapshot is critical; if the snapshot volume runs out of space, the backup will not be consistent.

All MySQL database files (data, log, indexes) must be stored in LVM logical volumes to ensure consistency. If any of the files are not on LVM, the snapshot is skipped, and either a raw backup via mysqlhotcopy or a logical backup using mysqldump will be taken based on the storage engines of the tables in each of the databases.

Configuration Parameters

The **Backup How** page allows you to select LVM snapshots as a backup mechanism.



Snapshot Size

Set the size of the LVM snapshot. For raw backups, each specified database is first checked to ensure that it is on an LVM volume, and then a snapshot of the specified size is created and used to backup the database (unless the quick (no-copy) option is selected; see below). If the specified database is not on a LVM volume, either **mysqlhotcopy** or **mysqldump** is used to create the backup.

Size of LVM snapshot depends on amount of activity in the logical volume during the backup window. This is difficult to predict. If the value is too small, the backup will fail. Select a value conservatively for the first backup run. The ZRM logs on the server (/var/log/mysql-zrm/mysql-zrm.log) shows the amount of snapshot space that was used during backup window when the backup completes successfully. This value can be used to tune the snapshot size configuration. Look for the value of **COW-table size** as shown below in the log message:

```
Tue May 04 12:59:28 2010: INFO: Output of the command sudo lvdisplay
/dev/nik_vg/zrm5pEeycW9LA 2>/tmp/ZRMKLOSo2o9 is   --- Logical volume ---
   LV Name                /dev/nik_vg/zrm5pEeycW9LA
   VG Name                nik_vg
   LV UUID                DronVf-GybO-rSQf-3Uqb-RG6I-krvP-aLTw8o
   LV Write Access        read/write
   LV snapshot status     active destination for /dev/nik_vg/lv_mysql
   LV Status              available
   # open                 0
   LV Size                30.00 GB
   Current LE             7680
   COW-table size         12.00 MB
   COW-table LE           3
   Allocated to snapshot  0.65%
   Snapshot chunk size    8.00 KB
   Segments               1
   Allocation             inherit
   Read ahead sectors     0
   Block device           253:4
```

Backup Type

Choose the method of snapshot backup. The Standard (Copy) option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The Quick (No Copy) option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the quick option is specified, the compress and encrypt options are ignored, and the backup size will always be zero. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the Convert Backup option available from the Reports menu tab.

Advantages of Using LVM Snapshots for Backup

- Hot backup for transaction based storage engines (no impact on the application using the database) and Warm backups for other storage engines.
- Backup time is not dependent on the size of the database. As a result, this backup method is suitable for large databases.
- Almost instantaneous. The database gets locked only for the time taken to create the snapshot.

Disadvantage of Using LVM Snapshots for Backup

- Works well only for filesystems that support freeze operation such XFS, VxFS.
- Additional disk space for logical volume snapshots is required.
- LVM snapshots can be used only for local backups.

Symantec VxFS snapshots

Symantec Veritas file systems (a.k.a VxFSs) allow two technologies to minimize application downtime during backup runs:

- *Snapshot devices,* which mirror the original storage and allow "freezing" for backup while the original data stays live.
- *Storage checkpoints,* which use the same filesystem volumes as the original to only mirror changed data.

ZRM for MySQL includes an optional VxFS plugin that integrates with the Veritas File System to leverage native Storage Checkpoint capability

This page describes the configuration of VxFS snapshot/storage checkpoint backups, including requirements for the MySQL database.

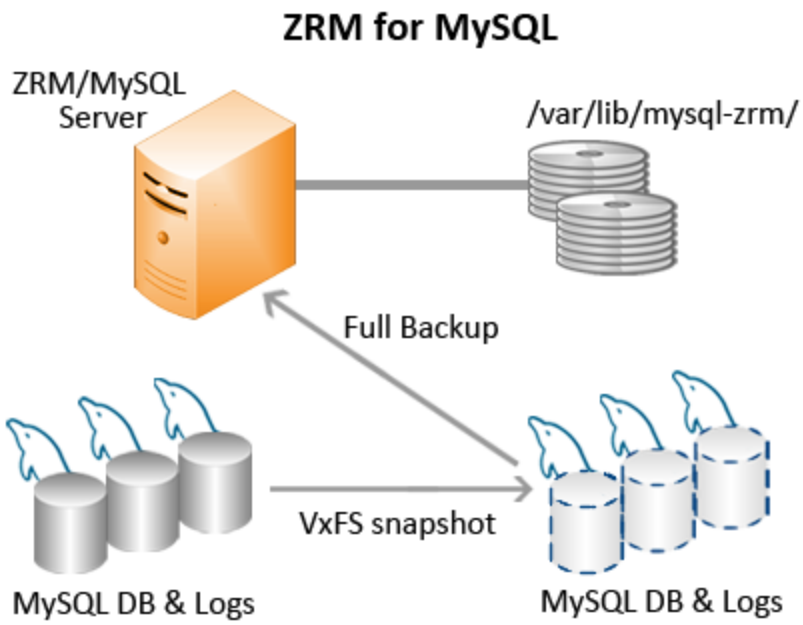
MySQL Configuration Requirements

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see the next section) On Solaris platforms, this means the **SMCsudo** package must be installed.
- If you are backing up a remote MySQL server, the destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

All MySQL data and logs must reside on VxFS volumes. The following are some of the possible configurations:

- All MySQL data is on a single VxFS Volume
- Specific database directories are on different volumes
- For databases containing InnoDB-based tables, the snapshot can only be used if the database directory, the InnoDB data files and the InnoDB logs are all on VxFS volumes.
    - ? The InnoDB shared data files are on a separate VxFS volume
    - ? The InnoDB logs are on a separate VxFS volume

Refer to Veritas-supplied VxFS documentation for details on VxFS storage checkpoint configuration.



The VxFS volumes are mounted on ZRM server. ZRM user "mysql" should have permissions to read and write to the volumes.

Pre-conditions for Using VxFS Snapshots

- The MySQL backup user must be granted **sudo** privileges to execute **VxFS** commands on the MySQL server. Add a line similar to the following example to **/etc/sudoers** on the MySQL server:

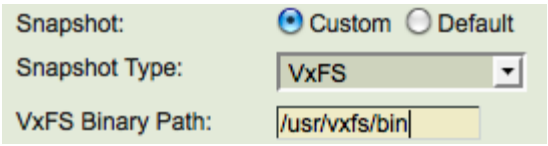
```
mysql MySQLserver.mycompany.com Server>=NOPASSWD:/bin/mount,NOPASSWD:/bin/umount
      NOPASSWD:/sbin/fsckptadm
```

  where *MySQLserver.mycompany.com* is the fully-qualified domain name for the MySQL server. Note that if **VxFS** commands are installed in non-standard locations, the above example would not work without editing it to reflect the different paths.

- All MySQL database files (data, log, indexes) must be stored in VxFS volumes to ensure consistency. If any of the files are not on VxFS volumes, a raw backup using mysqlhotcopy, or a logical backup using mysqldump will be taken based on the storage engines of the tables in each of the databases.

Configuring ZRM for MySQL to use VxFS Snapshots

To activate use of VxFS storage checkpoints, you must select the **VxFS** Snapshot Plugin from the **Backup How** page:



There are two options for VxFS snapshots:

VxFS Binary Path
Supply the path to VxFS **fsckptadm** command (the default is **/opt/VRTS/bin**

Backup Type
Choose the method of snapshot backup. The **Standard (Copy)** option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the **Convert Backup** option available from the **Reports** menu tab.

As noted, the backup mode (selected from the upper right of the **Backup How** page) must be **Raw.** If the current backup mode for the set is **Logical,** you cannot select any snapshot plugin from the **Backup How** page.

Network Appliance SnapManager

Network Appliance storage systems include the Network Appliance SnapManager software, which facilitates near-instantaneous hot backups and rapid restores.

ZRM for MySQL includes an optional snapshot plugin that integrates with SnapManager software to create consistent MySQL backups using ONTAP API. It creates temporary snapshots of the Network Appliance volumes to back up. The snapshots are removed when the backup run is completed. ZRM for MySQL using SnapManager performs backups with minimal impact on MySQL applications.

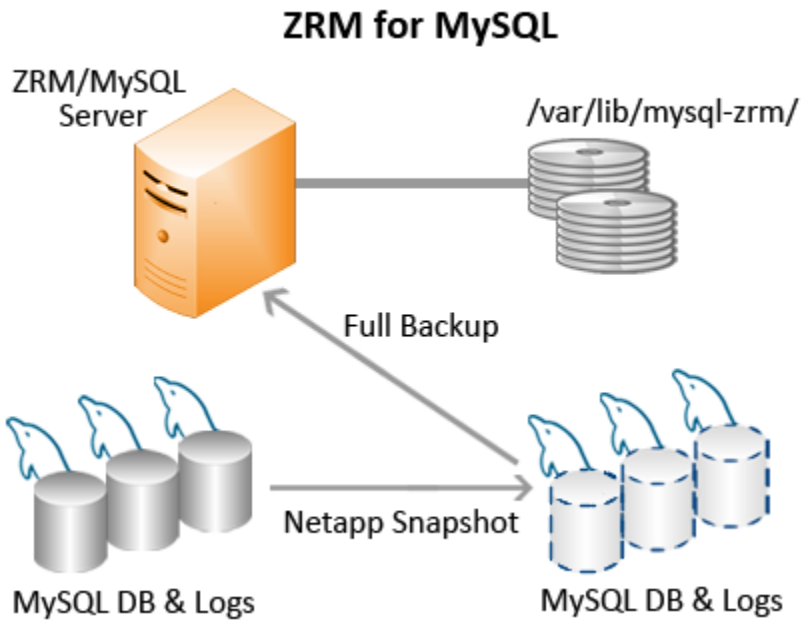
This page describes the command-line configuration of Network Appliance snapshot backups, including requirements for the MySQL database.

MySQL Configuration Requirements

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see below). On Solaris platforms, this means the **SMCsudo** package must be installed.
- If you are backing up a remote MySQL server, the destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

All MySQL data and logs must reside on Network Appliance volumes. The following are some of the possible configurations

- All MySQL data is on a single Network Appliance Volume
- Specific database directories are on different volumes
- For databases containing InnoDB-based tables, the snapshot can only be used if the database directory, the InnoDB data files and the InnoDB logs are all on Network Appliance volumes.
    - ? The InnoDB shared data files are on a separate Network Appliance volume
    - ? The InnoDB logs are on a separate Network Appliance volume



The Network Appliance volumes must be mounted on the ZRM server using NFS. The ZRM server's **mysql** user must have permissions to read and write to the volumes. If the Network Appliance volumes are accessed using some other protocol such as iSCSI, please contact Zmanda Support Team.

Pre-conditions for Using Network Appliance Snapshots

- The *mysql* operating system user must be granted **sudo** privileges to execute Network Appliance commands on the MySQL server. Add a line similar to the following example to **/usr/local/etc/sudoers** on the MySQL server:

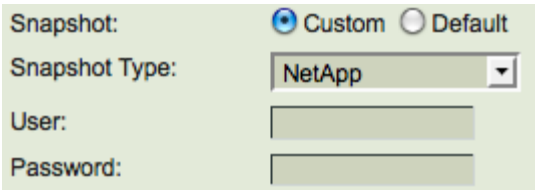
```
mysql MySQLserver.mycompany.com=NOPASSWD:/usr/sbin/mount,NOPASSWD:/usr/sbin/umou
```

where *MySQLserver.mycompany.com* is the fully-qualified domain name of the MySQL server. Note that if the *mount*, *umount* and *df* commands are installed in non-standard locations, the above example would not work without editing it to reflect the different paths.

- The snap reserve for the Network Appliance volume must be adequate to accommodate changed data. The Network Appliance default of 20% may be low or high for your site, depending on data change rates.
- To determine the snap reserve, use the Network Appliance **snap reserve** *volumename* command.
- All MySQL database files (data, log, indexes) must be stored in Network Appliance volumes to ensure consistency.
- If any of the files are not on Network Appliance volumes, a raw backup using mysqlhotcopy, or a logical backup using mysqldump will be taken based on the storage engines of the tables in each of the databases.
- If you using Netapp snapshots from a remote MySQL server (ZRM client installation), you will have to install Perl LWP-UserAgent module in the default perl installation.

Configuring ZRM for MySQL to use ZFS SnapManager

To activate use of Network Appliance Snapshots, you must select the **NetApp** SnapShot Plugin from the **Backup How** page:



NetApp User/Password

Supply a valid Network Appliance username and password. The Netapp user should have privileges to perform snapshots on the appliance.

Backup Type

Choose the method of snapshot backup. The **Standard (Copy)** option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the **Convert Backup** option available from the **Reports** menu tab.

As noted, the backup mode (selected from the upper right of the Backup How page) must be **Raw.** If the current backup mode for the set is **Logical,** you cannot select a Snapshot plugin from the Backup How page.

ZFS Snapshots

Sun Microsystem's Solaris ZFS filesystem include snapshot capability, which facilitates near-instantaneous hot backups and rapid restores.

If you purchase the feature license from Zmanda (available at the Zmanda Network Downloads page), ZRM for MySQL includes an optional snapshot plugin that integrates with ZFS to create consistent MySQL full backups. It creates temporary snapshots of the ZFS volumes on which to perform a full backup. When snapshots are enabled, ZRM for MySQL can perform backups with minimal impact on MySQL applications. Database writes will be blocked only during snapshot creation, which typically takes less than a second regardless of database size.

This page describes the configuration of ZFS snapshot backups, including requirements for the MySQL database.

MySQL Configuration Requirements

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see the next section). On Solaris platforms, this means the **SMCsudo** package must be installed.
- If you are backing up a remote MySQL server, the destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

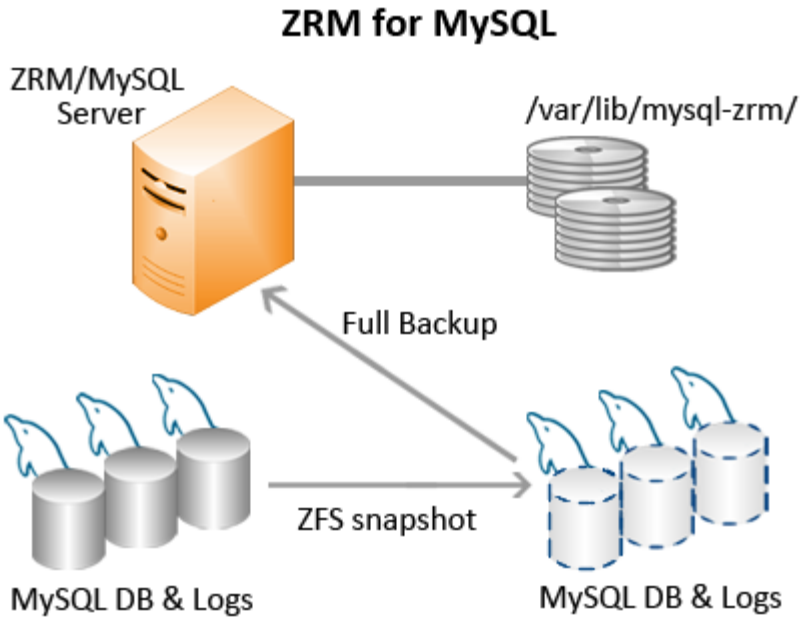
To take advantage of ZFS snapshots, all MySQL database files (data, log, indexes) belonging to the backup set must be stored in ZFS volumes to ensure consistency.

If any of the files are not on ZFS volumes, a raw backup using mysqlhotcopy, or a logical backup using mysqldump is performed depending on the storage engines of the tables in each of the databases.

Refer to the Solaris ZFS documentation for details on ZFS administration.

Here are some valid scenarios of database storage on ZFS:

- All MySQL data is stored on a single ZFS Volume
- Specific database directories are stored on different volumes
- For databases containing InnoDB-based tables, the snapshot can only be used if the database directory, the InnoDB data files and the InnoDB logs are all on ZFS volumes.
    - ? The InnoDB shared data files are on a separate ZFS volume
    - ? The InnoDB logs are on a separate ZFS volume



MySQL Backup User sudo Privileges

The MySQL backup user (described in System Requirements) must be granted **sudo** privileges to execute **ZFS** commands on the MySQL server. Add a line similar to the following example to **/usr/local/etc/sudoers** on the MySQL server:

```
mysql MySQLserver.mycompany.com=NOPASSWD:/usr/sbin/df,NOPASSWD:/usr/sbin/zfs
```

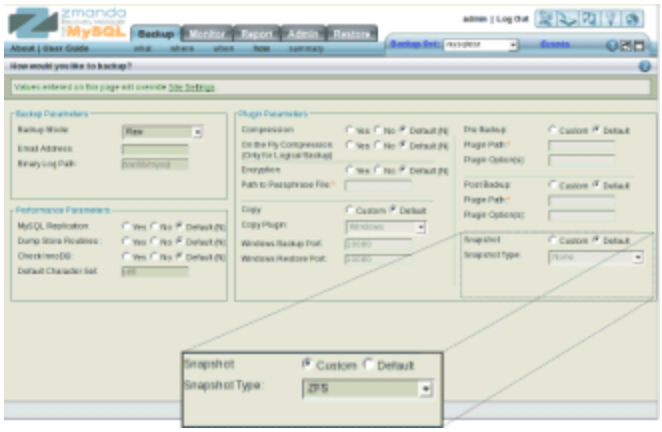
where *MySQLserver.mycompany.com* is the fully-qualified domain name for the MySQL server. If the MySQL database resides on the the ZRM server, the ZRM server name should be used. Note that if **ZFS** commands are installed in non-standard locations, the above example would not work without editing it to reflect the different paths.

To test the **sudo** configuration, run the command as the "mysql" user. The command should execute correctly without prompting for a password. For example:

```
# su - mysql
$ /usr/local/bin/sudo /usr/sbin/df
```

Configuring ZRM for MySQL to use ZFS Snapshots

To activate use of ZFS Snapshots, simply select the **ZFS** Snapshot Plugin from the **Backup How** page:



Backup Type

Choose the method of snapshot backup. The **Standard (Copy)** option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the **Convert Backup** option available from the **Reports** menu tab.

As noted, the backup mode (selected from the upper right of the **Backup How** page) must be **Raw.** If the current backup mode for the set is **Logical,** you cannot select a Snapshot plugin from the **Backup How** page.

EMC SnapView Snapshots

EMC Clariion Storage features EMC SnapView, a snapshot technology that facilitates near-instantaneous hot backups and rapid restores.

If you purchase the feature license from Zmanda (available at the Zmanda Network Downloads page), ZRM for MySQL includes an optional snapshot plugin that integrates with EMC SnapView to create consistent MySQL full backups. It creates temporary snapshots of the EMC SnapView volumes on which to perform a full backup. When snapshots are enabled, ZRM for MySQL can perform backups with minimal impact on MySQL applications. Database writes will be blocked only during snapshot creation, which typically takes less than a second regardless of database size.

This page describes the configuration of EMC SnapView snapshot backups, including requirements for the MySQL database.

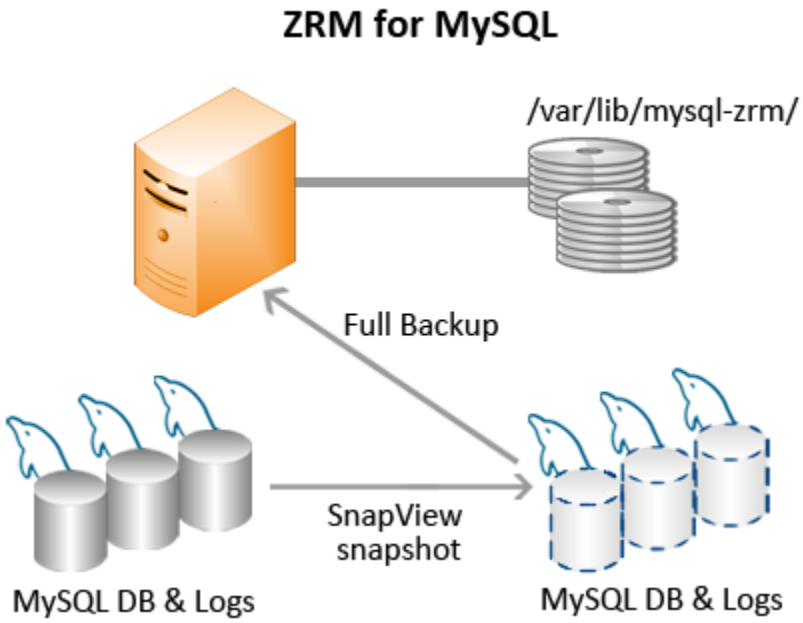
MySQL Configuration Requirements

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see the next section) On Solaris platforms, this means the **SMCsudo** package must be installed.

- If you are backing up a remote MySQL server, the destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

All MySQL data and logs must reside on Clariion storage.

Refer to EMC-supplied Clariion documentation for details on EMC SnapView .



The Clariion volumes are mounted on ZRM server (direct attached or SAN attached). ZRM user "mysql" should have permissions to read and write to the volumes.

Pre-conditions for Using EMC Snapview Snapshots

- The MySQL backup user must be granted **sudo** privileges to execute **EMC** commands on the MySQL server. Add a line similar to the following example to **/etc/sudoers** on the MySQL server:

```
mysql MySQLserver.mycompany.com Server>=NOPASSWD:/bin/mount,NOPASSWD:/bin/umount
      NOPASSWD:/opt/Navisphere/navicli
```

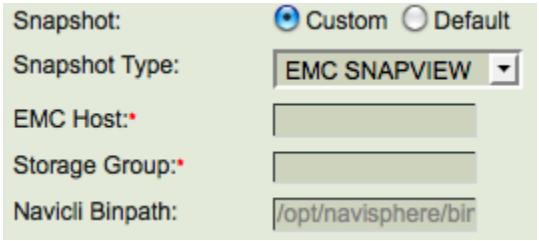
where *MySQLserver.mycompany.com* is the fully-qualified domain name for the MySQL server. Note that if **EMC** commands are installed in non-standard locations, the above example would not work without editing it to reflect the different paths.

- All MySQL database files (data, log, indexes) must be stored in SnapView volumes to ensure consistency. If any of the files are not on SnapView volumes, a raw backup using mysqlhotcopy, or a logical backup using mysqldump will be taken based on the storage engines of the tables in each of the databases.

- EMC NaviSphere (including the **navicli** and **naviseccli** utility) is required, and must be installed on the MySQL server being backed up and the ZRM server. This software is not shipped with ZRM for MySQL; contact EMC to obtain these programs.

Configuring ZRM for MySQL to use EMC SnapView Snapshots

To activate use of EMC SnapView Snapshots, simply select the **EMC SnapView** Snapshot Plugin from the **Backup How** page:



EMC Host
Specify the fully qualified domain name or IP address of the Clariion storage system. This is a mandatory parameter as there is no default.

Storage Group
Specify the EMC/Clariion Storage Group that you want to add to the snapshot. This is a mandatory parameter as there is no default.

Navicli Binpath
The path to where **navicli** and **naviseccli** are stored. The default value is **/opt/Navisphere/bin**.

Backup Type
Choose the method of snapshot backup. The **Standard (Copy)** option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the **Convert Backup** option available from the **Reports** menu tab.

As noted, the backup mode (selected from the upper right of the **Backup How** page) must be **Raw.** If the current backup mode for the set is **Logical,** you cannot select a Snapshot plugin from the **Backup How** page.

Bluearc Titan Snapshots

Bluearc Titan storage series provides snapshot capability that allows for near-instantaneous hot backups and rapid restores. BlueArc can store up to 1,024 snapshot images per volume on Titan.

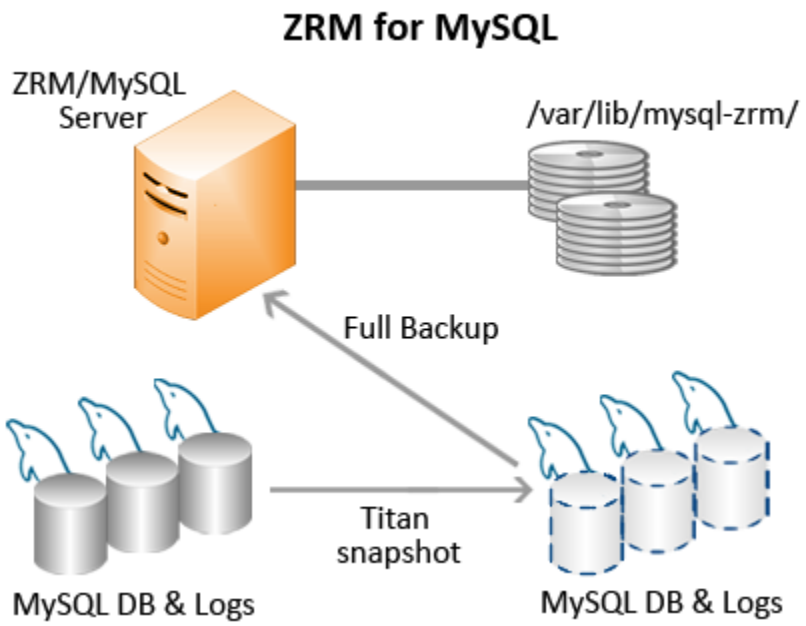
If you purchase the feature license from Zmanda (available at the Zmanda Network Downloads page), ZRM for MySQL includes an optional snapshot plugin that integrates with Bluearc Titan series to create consistent MySQL full backups. It creates temporary snapshots of the Bluearc volumes on which to perform a full backup. When snapshots are enabled, ZRM for MySQL can perform backups with minimal impact on MySQL applications. Database writes will be blocked only during snapshot creation, which typically takes less than a second regardless of database size.

This page describes the configuration of Bluearc Titan snapshot backups, including requirements for the MySQL database.

MySQL Configuration Requirements

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see the next section) On Solaris platforms, this means the **SMCsudo** package must be installed.
- If you are backing up a remote MySQL server, the destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

All MySQL data and logs must reside on Bluearc storage. Please see Bluearc Administration Guide on how to configure Bluearc storage volumes.



ZRM user "mysql" should have permissions to read and write to the volumes.

Pre-conditions for Using Bluearc Titan Snapshots

- The MySQL backup user must be granted **sudo** privileges to execute Bluearc **ssc** command and some system commands on the MySQL server. Add a line similar to the following example to **/etc/sudoers** on the MySQL server:

```
mysql MySQLserver.mycompany.com=NOPASSWD:/bin/mount,NOPASSWD:/bin/umount,NOPASSW
      NOPASSWD:<path to ssc command>
```

where *MySQLserver.mycompany.com* is the fully-qualified domain name for the MySQL server. Please correct the path to *ssc* command in the above example.

- All MySQL database files (data, log, indexes) must be stored in Bluearc storage to ensure consistency. If any of the files are not on Bluearc storage, a raw backup using mysqlhotcopy, or a logical backup using mysqldump will be taken based on the storage engines of the tables in each of the databases.
- The Bluearc *ssc* command is required on the MySQL server. This command is available from Bluearc for all customers. Please contact Bluearc for more information.
- BlueArc Virtual Volumes are logical entities that reside on top of the base volume of Titan. Snapshot of Virtual volumes is not supported. So, the base volumes will have to be exported to the MySQL server so that snapshot backup method can be used.
- The export name should be the same name as file system name i.e, File system label and Export name must be the same. For example: Consider the following Titan configuration

```
smet3-1:$ filesystem-list
Name                Dev   On span             State  EVS  Cap(GB)  Confined Flag
-----------------  ----  -----------------  -----  ---  -------  -------- ----
test1                1025  zManda              Mount   2      106        0

smet3-1:$ vn 2 nfs-export list | more
          Export name: /test1
          Export path: /
    File system label: test1
     File system size: 107 GB
File system free space: 19.0 GB
    File system state:
            formatted = Yes
              mounted = Yes
               failed = No
     thin provisioned = No
     Browse snapshots: Yes
         Read Caching: Disabled
```

Configuring ZRM for MySQL to use Bluearc Titan Snapshots

To activate use of Bluearc Titan Snapshots, simply select the **Bluearc Titan** Snapshot Plugin from the **Backup How** page. This option will appear only if the Zmanda optional license for Bluearc Titan snapshots has been installed on the ZRM server.

Bluearc IP address
IP address of Bluearc Titan storage device (hostnames are not allowed). This is a mandatory parameter as there is no default.

Username
User on the Bluearc storage. The user must be have Supervisor role. Supervisor can only perform snapshot operations. This paramater is mandatory.

Password
Bluearc storage user password.

SSC Binpath
The absolute path to *ssc* command. The *ssc* is command line tool available from Bluearc for Linux/Solaris platform. This command must be installed on the MySQL server.

Backup Type
Choose the method of snapshot backup. The **Standard (Copy)** option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the **Convert Backup** option available from the **Reports** menu tab.

As noted, the backup mode (selected from the upper right of the **Backup How** page) must be **Raw.** If the current backup mode for the set is **Logical,** you cannot select a Snapshot plugin from the **Backup How** page.

Amazon EC2 Cloud

ZRM for MySQL can backup MySQL servers running on Amazon EC2 instances in Amazon cloud. The backups are stored in Amazon S3 cloud storage. The backup images can be restored only to an Amazon EC2 instance. The backups are performed using Elastic Block Store (EBS) snapshots to S3.

If you purchase the feature license from Zmanda (available at the Zmanda Network Downloads page), ZRM for MySQL includes an optional snapshot plugin that integrates with Amazon EBS to create consistent MySQL full backups. It creates snapshots of the EBS volumes on which to perform a full backup. When snapshots are enabled, ZRM for MySQL can perform backups with minimal impact on MySQL applications. Database writes will be blocked only during snapshot creation, which typically takes less than a second regardless of database size.

This page describes the configuration of Amazon EBS snapshot backups, including requirements for the MySQL databases in the cloud.

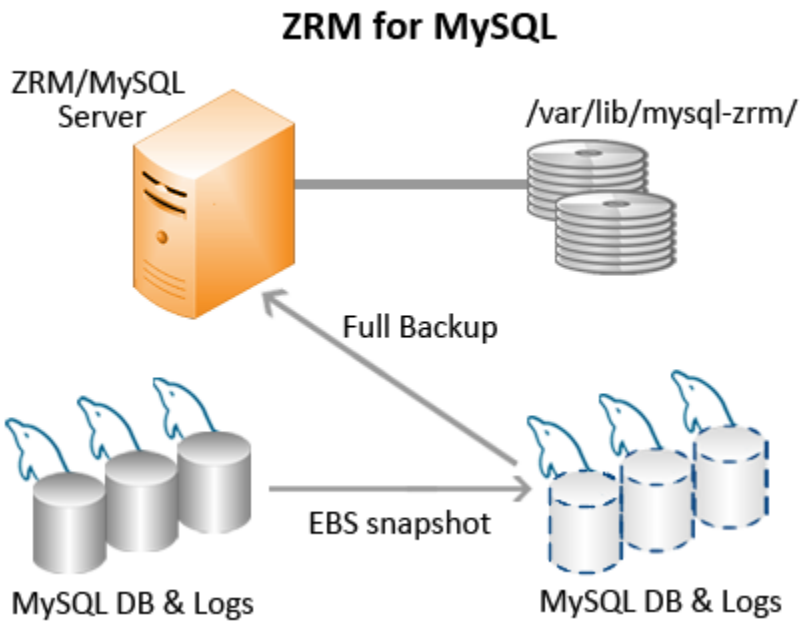
Requirements

- ZRM server and MySQL server must be running Linux EC2 instances. Solaris and Windows platforms are not supported.
- ZRM client software must be installed on the MySQL server EC2 instance.
- Perl module Net::Amazon::EC2 must be installed on the ZRM server under */opt/zmanda/zrm* directory. This perl module is available only from CPAN. To install this module, run the following commands as superuser on the ZRM server:

```
# /opt/zmanda/zrm/perl/bin/cpan

At CPAN prompt, type "install Net::Amazon::EC2"
```

- The destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.



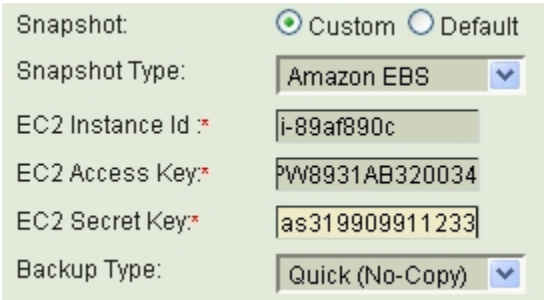
- All MySQL database files (data, log, indexes) must be stored in non-partitioned Amazon Elastic Block Store (EBS) volumes. ZRM user "mysql" should have permissions to read and write to the volumes. Please see Amazon documentation on how to use Elastic Block Storage in Amazon EC2.
- Only **Quick** backup method is supported. **Regular** backups cannot be performed.
- The number of EBS snapshots (full backups) that can be created depends on the Amazon account limit. The default limit is 500 EBS snapshots per amazon account. If full backups have to be retained for a long time, please contact Amazon to increase the EBS snapshot limit.
- Please make sure that the system date and time on the MySQL server and ZRM server if they are running on Amazon EC2 is correct. Incorrect date and time will result in failure of EBS snapshots and hence the backups.
- The MySQL backup user must be granted **sudo** privileges to execute system commands "df" and "xfs_freeze" (if XFS file system is in use) on the MySQL server. Add a line similar to the following example to **/etc/sudoers** on the MySQL server:

```
mysql ec2-75-101-206-181.compute-1.amazonaws.com=NOPASSWD:/bin/df,NOPASSWD:/usr/
```

where *ec2-75-101-206-181.compute-1.amazonaws.com* is the IP address of ZRM server on Amazon EC2.

Configuring ZRM for MySQL to use Amazon EBS Snapshots

- Install the EBS snapshot license on the ZRM server. You will have to download the license file from Zmanda network and copy it to */etc/zmanda* directory.
- Choose the Amazon EC2 instance's private DNS name as the host name for the MySQL server in the Backup|What page. For example: domU-12-31-39-06-70-22.compute-1.internal
- Select *raw* from the drop down menu for *Backup Mode* in Backup|How page in Zmanda Management Console for the backup set.
- Select *custom* radio button for Snapshot and select *Amazon EBS* in the Snapshot Type drop down box.



Amazon EC2 access key

Amazon account access key identifier. This information can be obtained from Amazon account page (Select Your Account -> Access Identifiers)

Amazon EC2 secret key
Amazon account secret key identifier. This information can be obtained from Amazon account page (Select Your Account -> Access Identifiers)

Amazon EC2 instance id
The Instance ID of the Amazon EC2 instance where MySQL server is running. This MySQL server should be configured in the Backup|What page in the backup set. Usually, the ID begins with string "i-". This information can be obtained from the AWS Management Console (Amazon account information is required).

Backup Type
Only **Quick** backup is supported. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores. Backups are stored in highly reliable and scalable Amazon S3 that offers protection against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups using EBS cannot be converted into **regular** backups.

Restoring EBS snapshot backup

Elastic Block store snapshot backups can be restored only to an Amazon EC2 instance. There are two ways to do restoration of EBS snapshot backups.

- Restoration of EBS snapshots to an EC2 where MySQL server is already running. This method is the recommended method. It can be used to restore MySQL backups back to original EC2 instance. This method is supported by Zmanda Management Console. For more details on how to configure destination Amazon EC2 instance id and other EBS parameters in the restore process, please see Restore Where page.
- Restoration of EBS snapshots to an EC2 where MySQL server is not running. This method is useful for quick recovery of MySQL data. This restoration option is available in command line only.

***mysql-zrm-manage-backup --mount-snapshots --source-directory <directory where backup images are stored> ***
***--ec2-instance-id <instance id of amazon ec2 where the snapshots have to be mounted> ***
--device-fs-map <name of the mapping file containing devices and mount points>

The EBS devices are mounted at the mount points specified in the device mapping file on the destination amazon ec2 instance. The user can start a MySQL server with the datadir and logdir pointing to the mounted EBS devices to start accessing restored data.

An example of device mapping file:

/dev/sdf=/db
/dev/sdk=/innodb_data
/dev/sdn=/innodb_logs

and MySQL server to use the above data will have to be configured as follows in the MySQL server options file (**my.cnf**)

datadir=/db
innodb_data_home_dir=/innodb_data
innodb_log_group_home_dir=/innodb_logs

Use *--dismount-existing* option to **mysql-zrm-manage-backup** command if the EC2 where backups are being restored to already has EBS volumes mounted at the mount points.

Network Appliance SnapManager

Network Appliance storage systems include the Network Appliance SnapManager software, which facilitates near-instantaneous hot backups and rapid restores.

ZRM for MySQL includes an optional snapshot plugin that integrates with SnapManager software to create consistent MySQL backups using ONTAP API. It creates temporary snapshots of the Network Appliance volumes to back up. The snapshots are removed when the backup run is completed. ZRM for MySQL using SnapManager performs backups with minimal impact on MySQL applications.

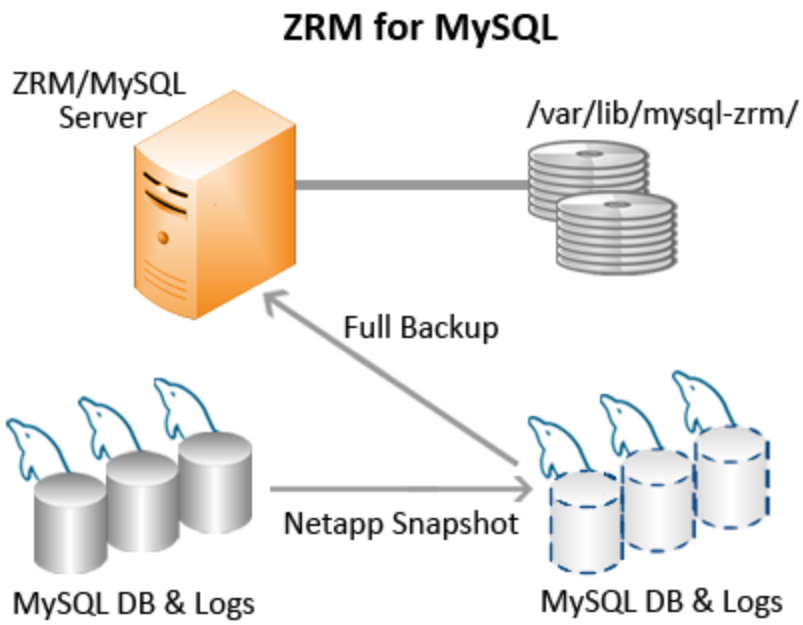
This page describes the command-line configuration of Network Appliance snapshot backups, including requirements for the MySQL database.

MySQL Configuration Requirements

- Both the ZRM server and MySQL server require the configuration of sudo privileges (see below). On Solaris platforms, this means the **SMCsudo** package must be installed.
- If you are backing up a remote MySQL server, the destination directory specified on the Backup Where page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS-level) must have read/write permission to access this directory.

All MySQL data and logs must reside on Network Appliance volumes. The following are some of the possible configurations

- All MySQL data is on a single Network Appliance Volume
- Specific database directories are on different volumes
- For databases containing InnoDB-based tables, the snapshot can only be used if the database directory, the InnoDB data files and the InnoDB logs are all on Network Appliance volumes.
    - ? The InnoDB shared data files are on a separate Network Appliance volume
    - ? The InnoDB logs are on a separate Network Appliance volume



The Network Appliance volumes must be mounted on the ZRM server using NFS. The ZRM server's **mysql** user must have permissions to read and write to the volumes. If the Network Appliance volumes are accessed using some other protocol such as iSCSI, please contact Zmanda Support Team.

Pre-conditions for Using Network Appliance Snapshots

- The *mysql* operating system user must be granted **sudo** privileges to execute Network Appliance commands on the MySQL server. Add a line similar to the following example to **/usr/local/etc/sudoers** on the MySQL server:

```
mysql MySQLserver.mycompany.com=NOPASSWD:/usr/sbin/mount,NOPASSWD:/usr/sbin/umou
```

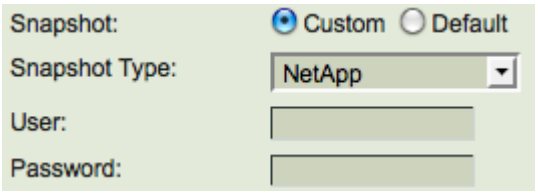
  where *MySQLserver.mycompany.com* is the fully-qualified domain name of the MySQL server. Note that if the *mount*, *umount* and *df* commands are installed in non-standard locations, the above example would not work without editing it to reflect the different paths.

- The snap reserve for the Network Appliance volume must be adequate to accommodate changed data. The Network Appliance default of 20% may be low or high for your site, depending on data change rates.
- To determine the snap reserve, use the Network Appliance **snap reserve** *volumename* command.
- All MySQL database files (data, log, indexes) must be stored in Network Appliance volumes to ensure consistency.

- If any of the files are not on Network Appliance volumes, a raw backup using mysqlhotcopy, or a logical backup using mysqldump will be taken based on the storage engines of the tables in each of the databases.
- If you using Netapp snapshots from a remote MySQL server (ZRM client installation), you will have to install Perl LWP-UserAgent module in the default perl installation.

Configuring ZRM for MySQL to use ZFS SnapManager

To activate use of Network Appliance Snapshots, you must select the **NetApp** SnapShot Plugin from the **Backup How** page:



NetApp User/Password

Supply a valid Network Appliance username and password. The Netapp user should have privileges to perform snapshots on the appliance.

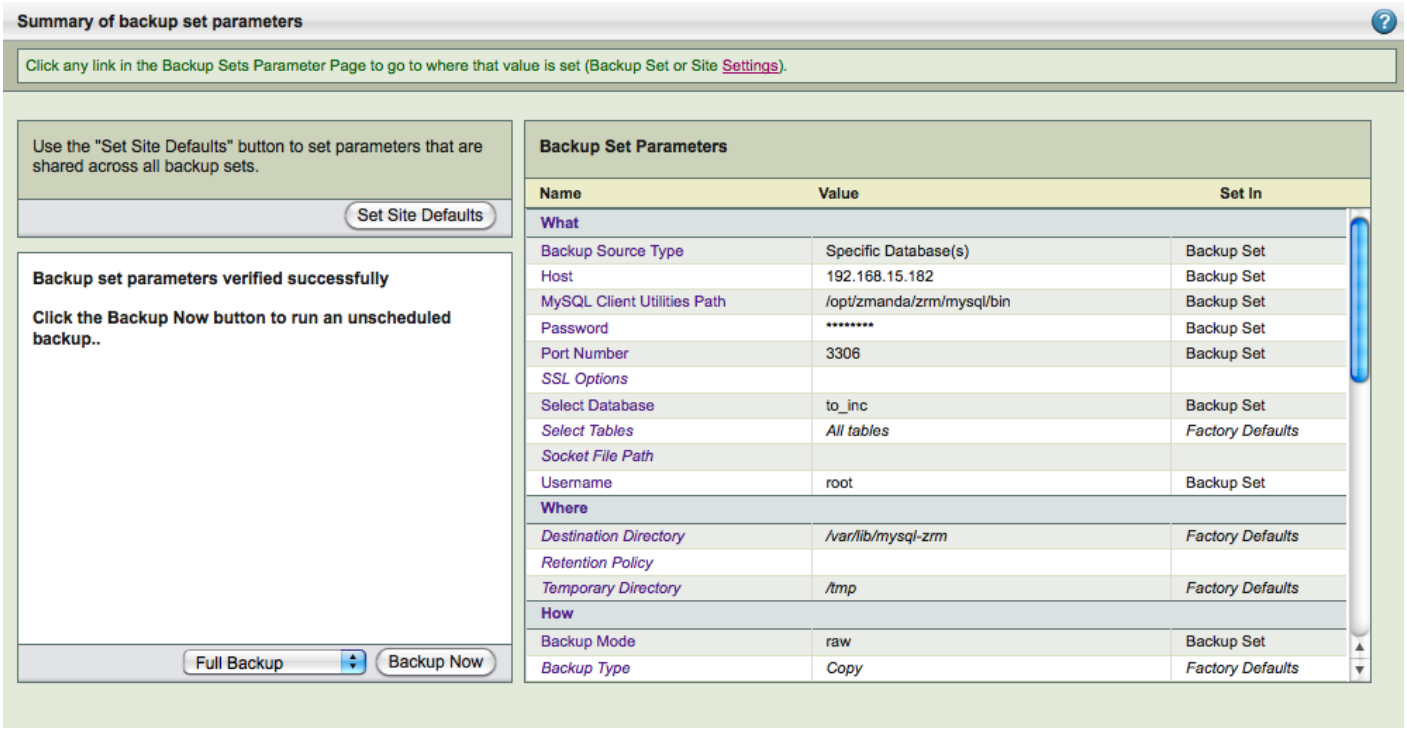
Backup Type

Choose the method of snapshot backup. The **Standard (Copy)** option specifies that the snapshot should be copied to a standard ZRM for MySQL backup archive. The **Quick (No Copy)** option specifies that the snapshot itself should be used as a near-line backup. Quick backups are convenient as they provide faster backups and restores, but because they remain on the MySQL server they do not protect against media or server failure. Note that if the **quick** option is specified, the **compress** and **encrypt** options are ignored. In addition, no checksums are performed, which means that quick snapshot backups cannot be verified. Quick snapshot backups may be converted to standard backups stored on the ZRM server using the **Convert Backup** option available from the **Reports** menu tab.

As noted, the backup mode (selected from the upper right of the Backup How page) must be **Raw.** If the current backup mode for the set is **Logical,** you cannot select a Snapshot plugin from the Backup How page.

The Backup Summary page conveniently lists all revelent settings for the backup set in one place, showing each option, and where the option was set (i.e., either factory, site, or in the backup set itself).

Backup Summary provides information about the configuration parameter values for the backup set and where it was set - **Factory Settings** (ZRM default value), **Site Defaults** (Set in **Set Site Defaults** page) and Backup Set (Set in Backup What/Where/When/How pages).



The summary items are listed under headings that correspond to the Backup page where the option is set. Note that the "Backup Type" field in this context refers to whether a copy of the backup was transferred to the ZRM server. This field will display **Copy** for all backup operations except for Quick Snapshots, where it will display **No-copy**.

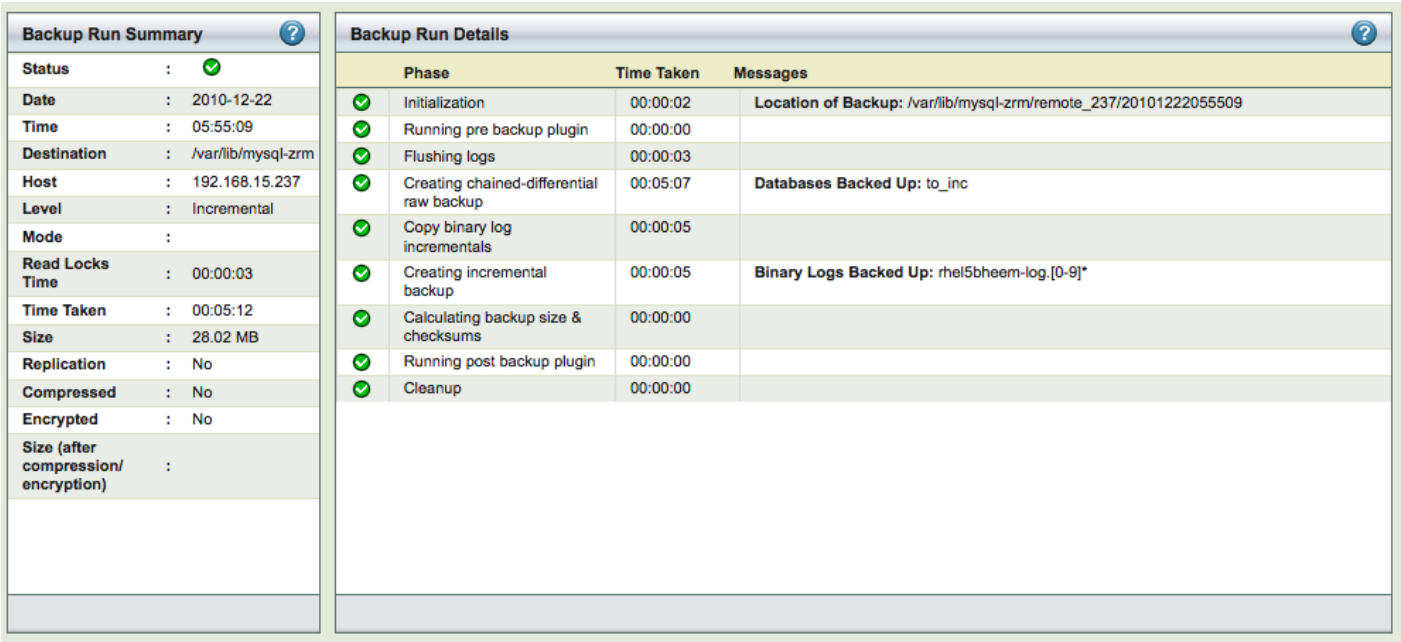
You can click on the **Set Site Defaults** to see and modify the site configuration default values for the ZRM server.

This page allow performs configuration check on ZRM server and the MySQL server for the backup set configuration. All errors/warnings are displayed in the left panel. If there are errors, backups will not be performed. Warnings about disk space availability on the backup destination location are also displayed. It is important to enable binary logs for the MySQL server for both full as well log incremental backups.

You can also start Full, Log Incremental, Differential, Chained Differential backups using **Backup Now** button. Last two backup options are available only if MySQL Enterprise Backup is selected in ZMC **Backup How** page. This is a good way to test the backups before scheduled backups run.

Monitoring Backups

The **Monitor** page shows the progress of backup run. The Backup run goes through various phases. Each phase is displayed and time taken by each phase is displayed when it is completed. It is possible to abort a backup run when it is absolutely necessary. The backup will stop at the next appropriate time (most likely at the beginning of the next phase). The cancel button will appear during the backup run. The client process are not terminated and will have to be terminated manually.



The left panel shows summary and configuration of the backup run. The Monitor shows information about the most recent backup run or the backup run in progress.

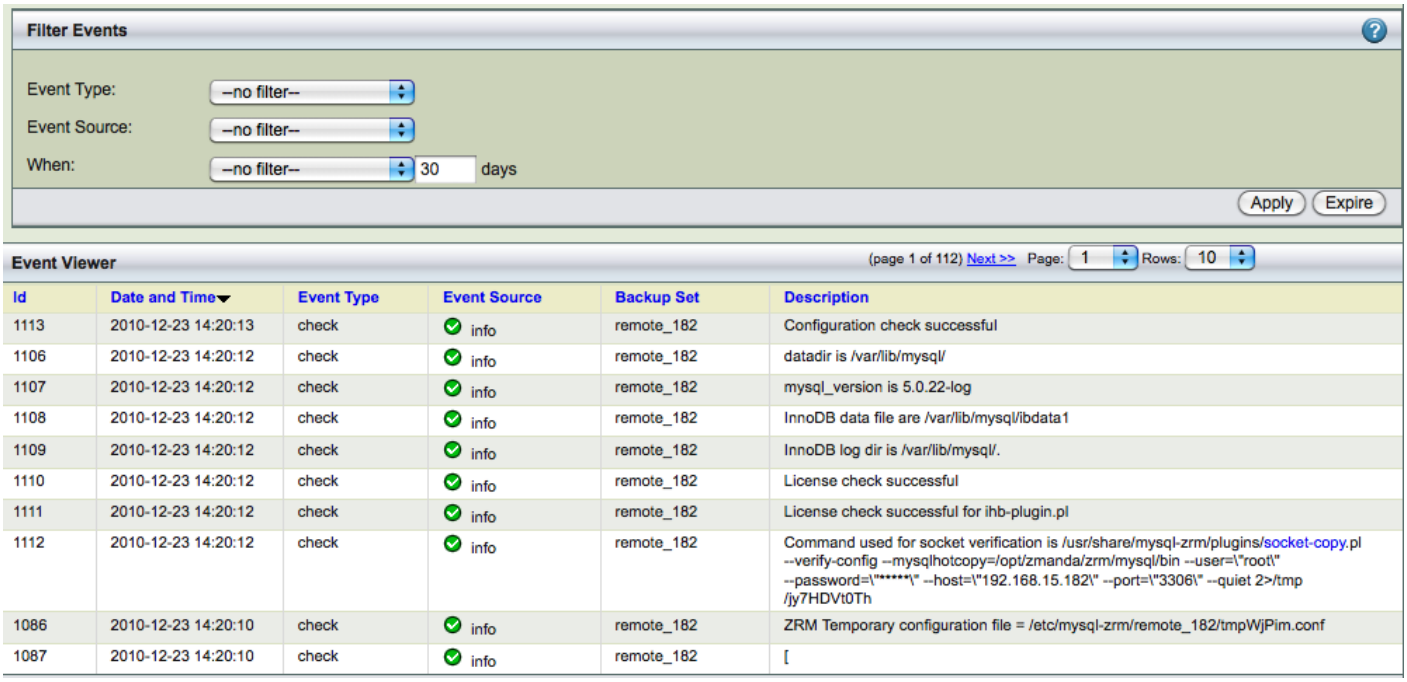
The backup processes divided into multiple phases. The right panel shows the progress of each phase. The name of the phase, time taken for each phase and any important messages regarding the phase are displayed.

Backup Status icons

- The yellow exclamation point (!) indicates warnings. They may require attention to prevent future backup failures.
- The green checkmark symbol indicates a task that executed successfully.
- The red octagon ("stop sign") symbol indicates backup has failed.

Events

ZMC provides all backup and configuration check events across backup sets in the **Event Viewer** as shown below.


Filter Events
Event Type: --no filter--
Event Source: --no filter--
When: --no filter-- 30 days
Apply Expire
Event Viewer
(page 1 of 112) Next >> Page: 1 Rows: 10
Id | Date and Time | Event Type | Event Source | Backup Set | Description
1113 | 2010-12-23 14:20:13 | check | info | remote_182 | Configuration check successful
1106 | 2010-12-23 14:20:12 | check | info | remote_182 | datadir is /var/lib/mysql/
1107 | 2010-12-23 14:20:12 | check | info | remote_182 | mysql_version is 5.0.22-log
1108 | 2010-12-23 14:20:12 | check | info | remote_182 | InnoDB data file are /var/lib/mysql/ibdata1
1109 | 2010-12-23 14:20:12 | check | info | remote_182 | InnoDB log dir is /var/lib/mysql/.
1110 | 2010-12-23 14:20:12 | check | info | remote_182 | License check successful
1111 | 2010-12-23 14:20:12 | check | info | remote_182 | License check successful for ihb-plugin.pl
1112 | 2010-12-23 14:20:12 | check | info | remote_182 | Command used for socket verification is /usr/share/mysql-zrm/plugins/socket-copy.pl --verify-config --mysqlhotcopy=/opt/zmanda/zrm/mysql/bin --user=\"root\" --password=\"******\" --host=\"192.168.15.182\" --port=\"3306\" --quiet 2>/tmp/jy7HDVt0Th
1086 | 2010-12-23 14:20:10 | check | info | remote_182 | ZRM Temporary configuration file = /etc/mysql-zrm/remote_182/tmpWjPim.conf
1087 | 2010-12-23 14:20:10 | check | info | remote_182 | [

Event Viewer

The columns of the page are:

Id

Event identifier

Date and Time

Time stamp of the event, displayed in 24-hour format

Event Type

Configuration **check** and **backup**

Event Source

The source of the event or log is either the local ZMC console or the Zmanda network. It also shows *failure*, *warning*, or *info*. All *failure* events need immediate attention.

Backup Set

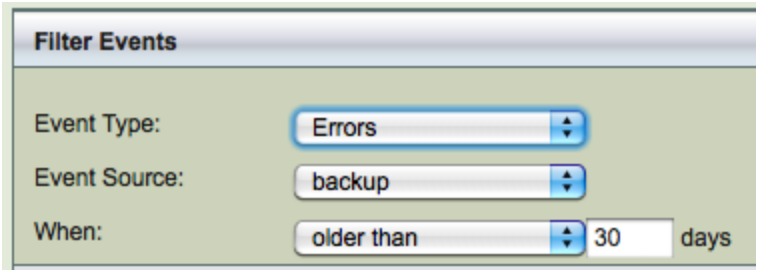
Displays the backup set name if the event is not a Zmanda Network Alert.

Description

Event description. Some event descriptions have links. Clicking the links will provide information from knowledgebase in the Zmanda Network or the wiki about how to resolve the problem if it is an error.

Alerts can also come from Zmanda Network, providing security and product updates. The ZMC can generate many events during a backup process and configuration process. By clicking on the header of any column, the view of the page can be altered so that it is sorted as per that column.

Filter Events



You can select the events in the viewer using Event Type and Event Source. Event Type can be Errors or Warnings. Event Source can be backup or check. You can filter based on the time stamp.

You can purge events using **Expire** button. Select the events to be removed using **When** field. Events take space in the internal database and if there are disk space constraints, it is important to expire events regularly when they are no longer needed.

Log Rotate Utility

ZMC works with a log rotate utility that allows sysadmins to effectively prune active logs.

Sysadmins should rotate the logs using **crontab**.

```
For example:
0 1 * * 1,5 logrotate /etc/logrotate.d/zmc_logrotate (For 1 AM on Monday and Fri
```

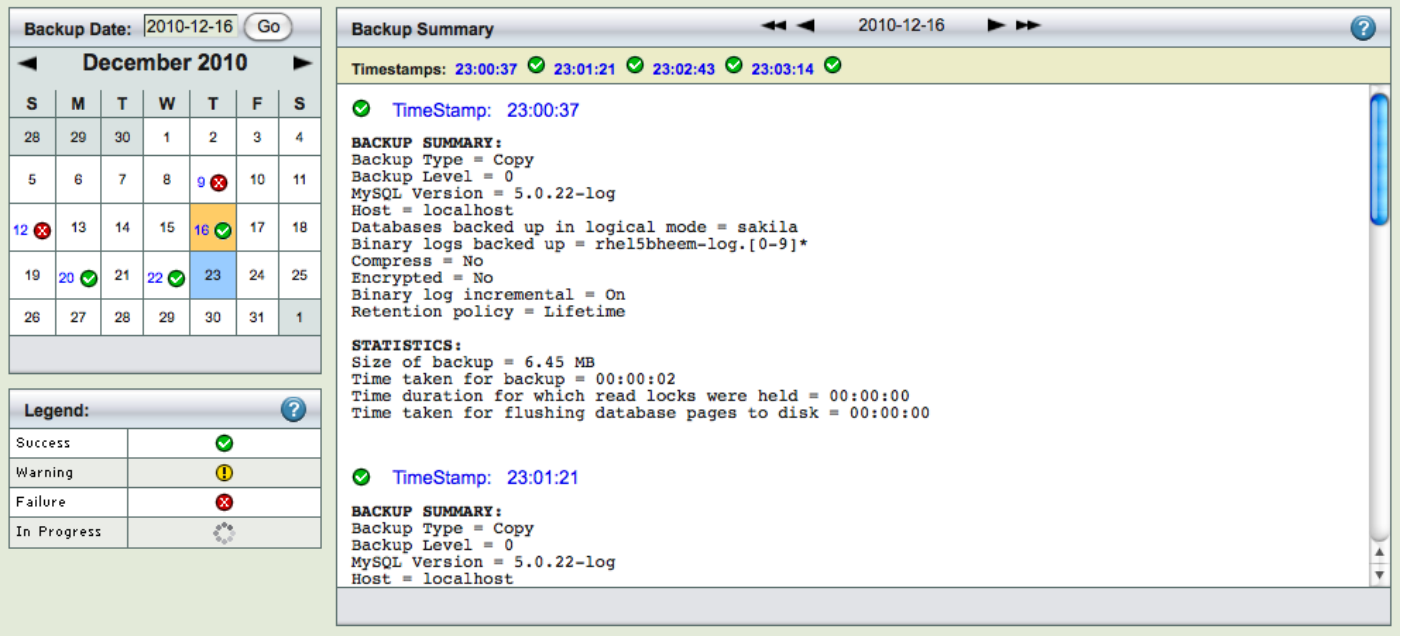
The pruned logs are not saved by the utility; they are simply pruned by it. You must manually copy the logs before pruning if you wish to retain them.

Summary Report

ZRM for MySQL automatically generates backup reports after the backup run is completed (or if it fails for some reason). This backup summary report includes:

- status of the current/last backup run
- the backup type (in other words, whether the backup was a quick (no-copy) snapshot backup.
- backup statistics
- location of the backup image
- backup level.

If the machine where ZMC is running is configured to send mail, reports can be automatically e-mailed after every backup run.


Backup Date: 2010-12-16 Go
December 2010
S M T W T F S
28 29 30 1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31 1
Legend:
Success
Warning
Failure
In Progress
Backup Summary
2010-12-16
Timestamps: 23:00:37 23:01:21 23:02:43 23:03:14
TimeStamp: 23:00:37
BACKUP SUMMARY:
Backup Type = Copy
Backup Level = 0
MySQL Version = 5.0.22-log
Host = localhost
Databases backed up in logical mode = sakila
Binary logs backed up = rhel5bheem-log.[0-9]*
Compress = No
Encrypted = No
Binary log incremental = On
Retention policy = Lifetime
STATISTICS:
Size of backup = 6.45 MB
Time taken for backup = 00:00:02
Time duration for which read locks were held = 00:00:00
Time taken for flushing database pages to disk = 00:00:00
TimeStamp: 23:01:21
BACKUP SUMMARY:
Backup Type = Copy
Backup Level = 0
MySQL Version = 5.0.22-log
Host = localhost

The Summary Report page is divided into two panels.

- The left panel shows a calendar control from which you can select report dates, and a legend that explains the report icons.
- The right panel displays the report for date selected on the calendar.

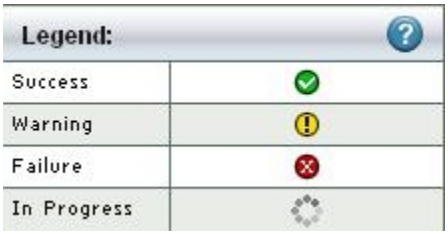
You can select reports using any of the following:

- Browse buttons at the top of the report itself.
- You can enter a date on the left panel and click the **Go** button.
- You can pick a date off the calendar.

Selecting a backup date

You can either enter the date (*mm*/*dd*/*yy*[*yy*]) and click **Go**, or click on any icon in the calendar. When you click **Go** (or click on a different date), the report shows any summary data (if any is available) for that date.

Legend



Legend shows four possible status for a backup run:

Error
Requires immediate attention

Warning
A recoverable failure

Success
Successful backup run, the data backed up can be recovered from this image

Progress
Backup run is in progress. Please check Monitor page for more information about this backup run

Summary Panel

Backup Summary
2008-04-17
Timestamps: 15:25:58 15:58:13 15:58:59
TimeStamp: 15:58:13
BACKUP SUMMARY:
Backup Type = No-Copy
Backup Level = 0
MySQL Version = 5.0.27-standard-log
Host = 192.168.15.143
Comment = testing quickback up
Databases backed up in raw mode using snapshot = test
InnoDB shared data files backed up = /var/lib/mysql/ibdata1;
InnoDB logs backed up = /var/lib/mysql/./ib_logfile*
Compress = No
Encrypted = No
STATISTICS:
Size of backup = 0.00 MB
Time taken for backup = 00:00:12
Time duration for which read locks were held = 00:00:03
Time taken for flushing database pages to disk = 00:00:00
TimeStamp: 15:58:59

The date browse buttons allow to conveniently move back and forth one day (using < or >) or one week (using << or>>) at a time.

Timestamp links below the browse buttons show the time at which the backup run was initiated. If there are multiple backups in a day, you will see multiple time stamps. Clicking on a Timestamp link go to the Restore What page with the date and time automatically filled in.

Backup Summary
2008-04-17
Timestamps: 15:25:58 15:58:13 15:58:59
TimeStamp: 15:58:13
BACKUP SUMMARY:
Backup Type = No-Copy
Backup Level = 0
MySQL Version = 5.0.27-standard-log
Host = 192.168.15.143
Comment = testing quickback up
Databases backed up in raw mode using snapshot = test
InnoDB shared data files backed up = /var/lib/mysql/ibdata1:
InnoDB logs backed up = /var/lib/mysql/./ib_logfile*
Compress = No
Encrypted = No
STATISTICS:
Size of backup = 0.00 MB
Time taken for backup = 00:00:12
Time duration for which read locks were held = 00:00:03
Time taken for flushing database pages to disk = 00:00:00
TimeStamp: 15:58:59

Backup Summary

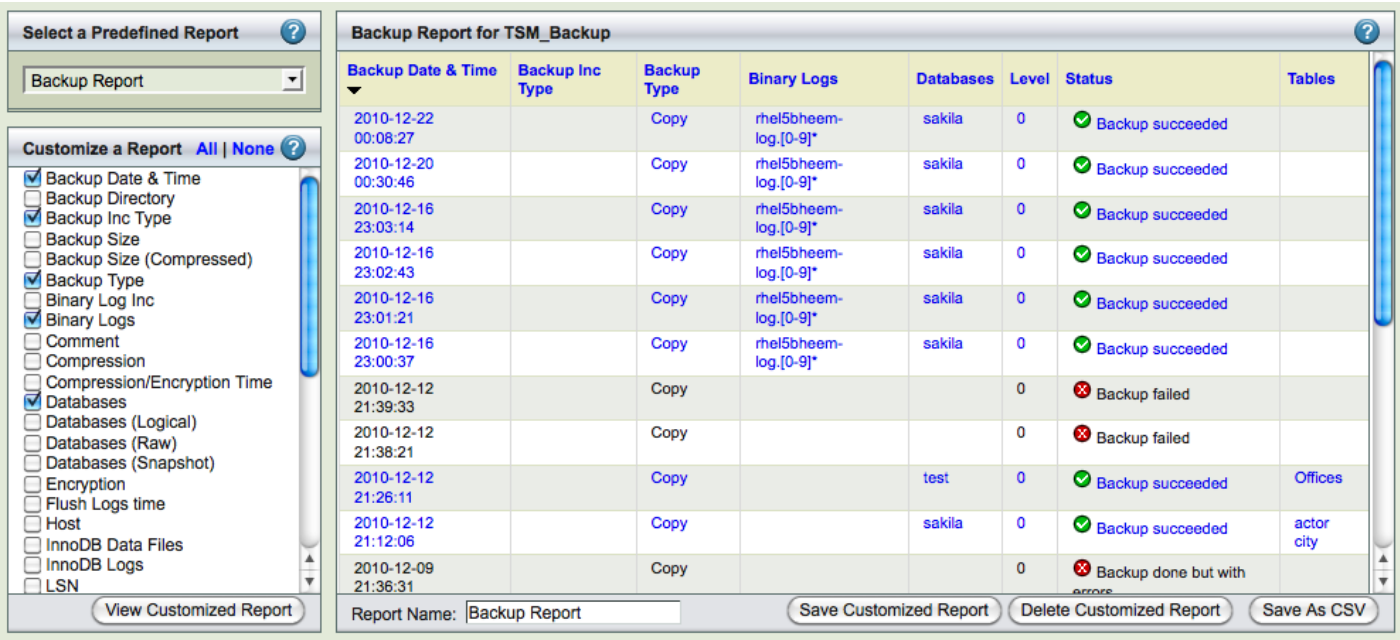
The what, when, where and how of the backup run. Note that **Backup Type** indicates whether the backup is a standard backup copied to the ZRM server (**Copy**), a quick snapshot backup (**No-copy**), or a quick snapshot that has been converted to a standard backup (**Copy (converted)**).

Statistics

How much data was backed up and how long it took. Note that quick snapshot (**No-copy**) backups always display a backup size of zero, because no data transfer occurs when this type of backup is run.

Custom reports

ZRM for MySQL generates a number of Predefined backup reports you can easily customize. ZRM for MySQL automatically generates backup reports after each backup run is completed.

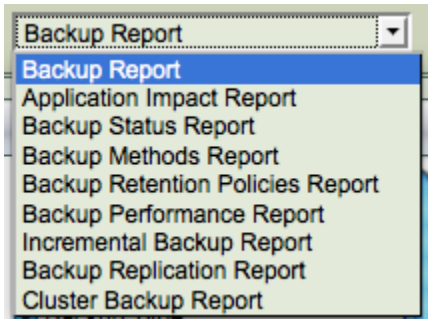


The Custom Reports displays a list of predefined reports, along with checkboxes that allow the user to customize reports.

- The left panel shows a list of predefined reports; beneath this list is a series of checkboxes that let you customize the display.
- The right panel displays the Report.

Predefined Reports

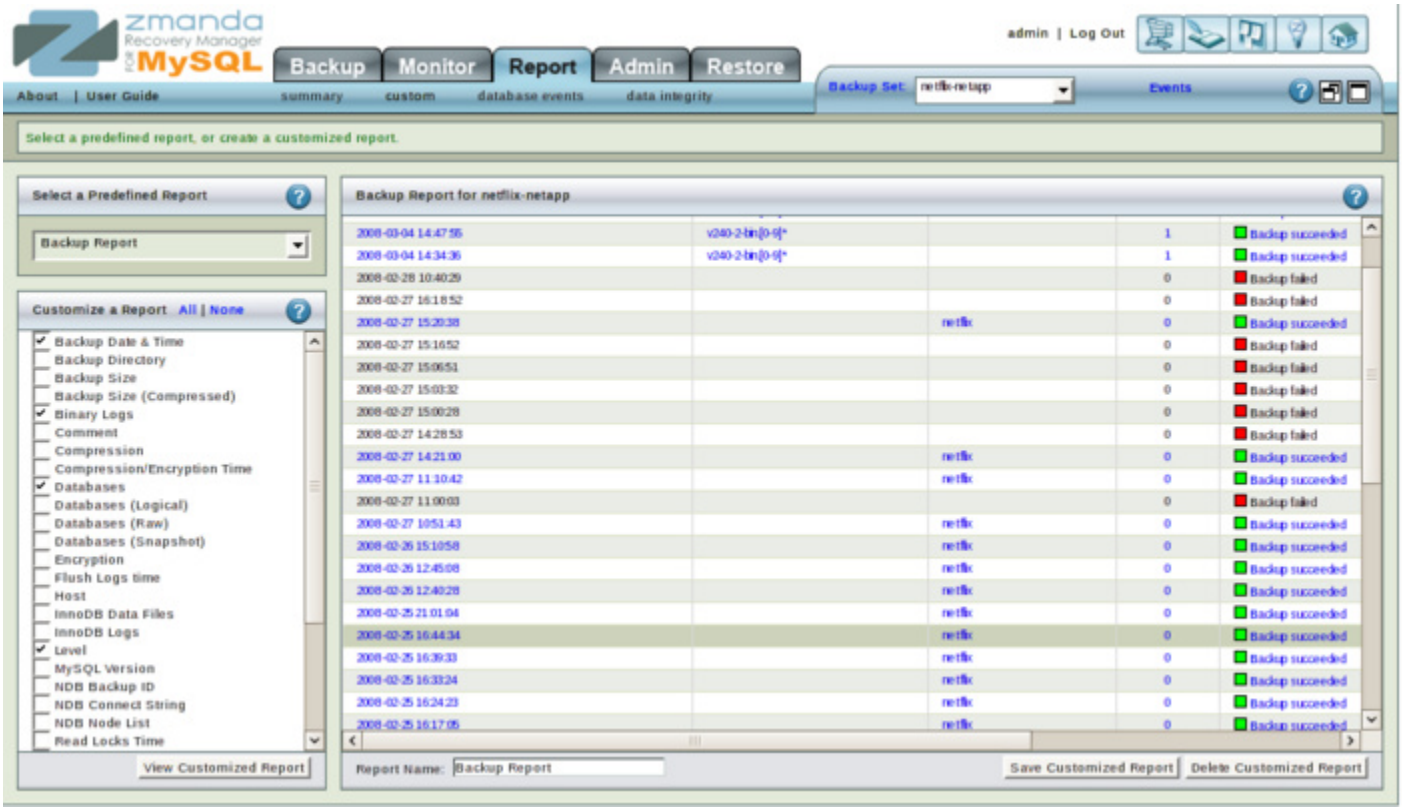
The list of Predefined Reports and User defined reports that can be run for the selected backup set can be selected from the drop down box on the left panel.
The predefined reports are shown and described below. The reports are shown displaying the default columns; you can customize each of the reports to display the desired



column fields. The Backup Date and Time column is common to all the reports. It identifies and differentiates between different Backup Runs of the same backup set. A few key columns are discussed below.

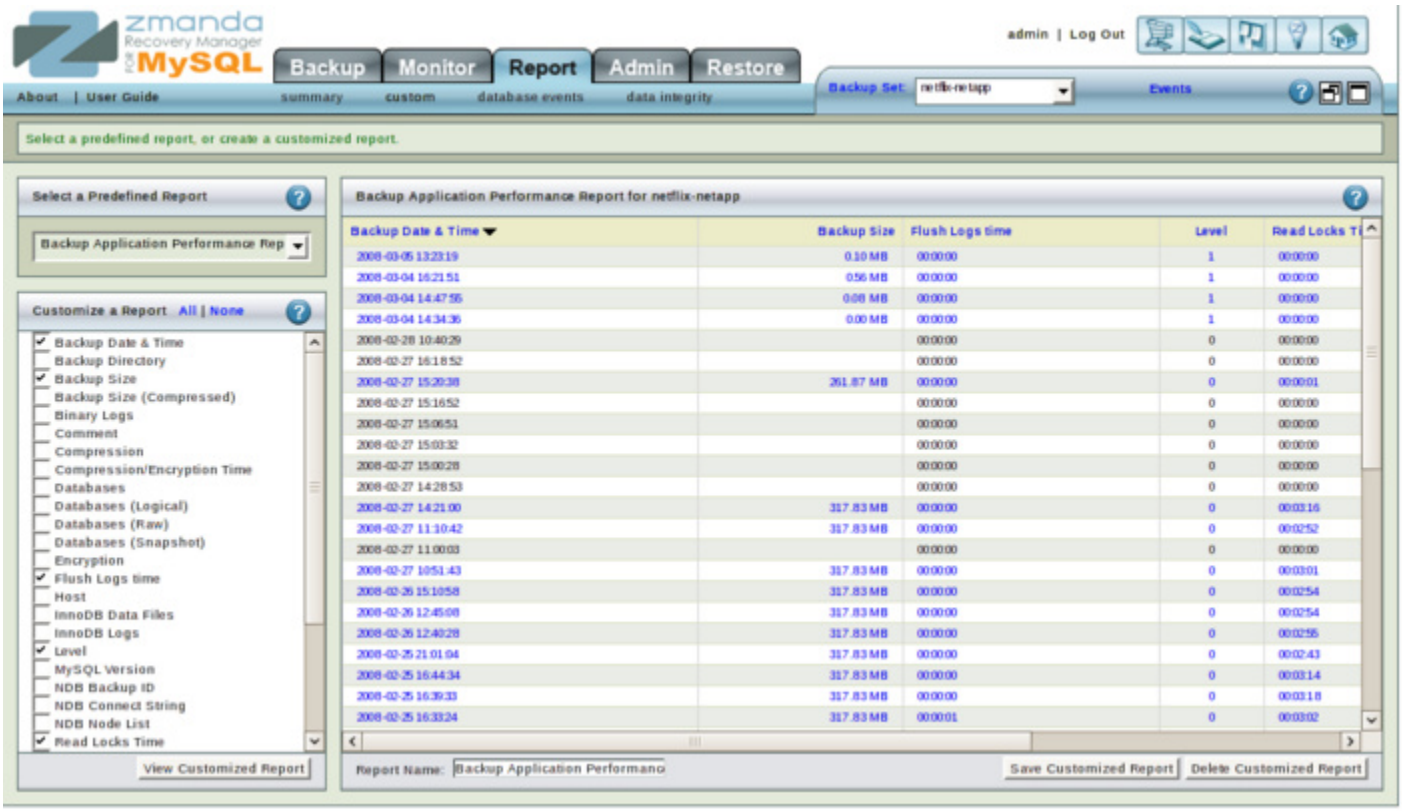
When any link is clicked, ZRM for MySQL jumps to the **Restore What** page with the date and time filled in based on the backup you selected from the report.

Backup Report



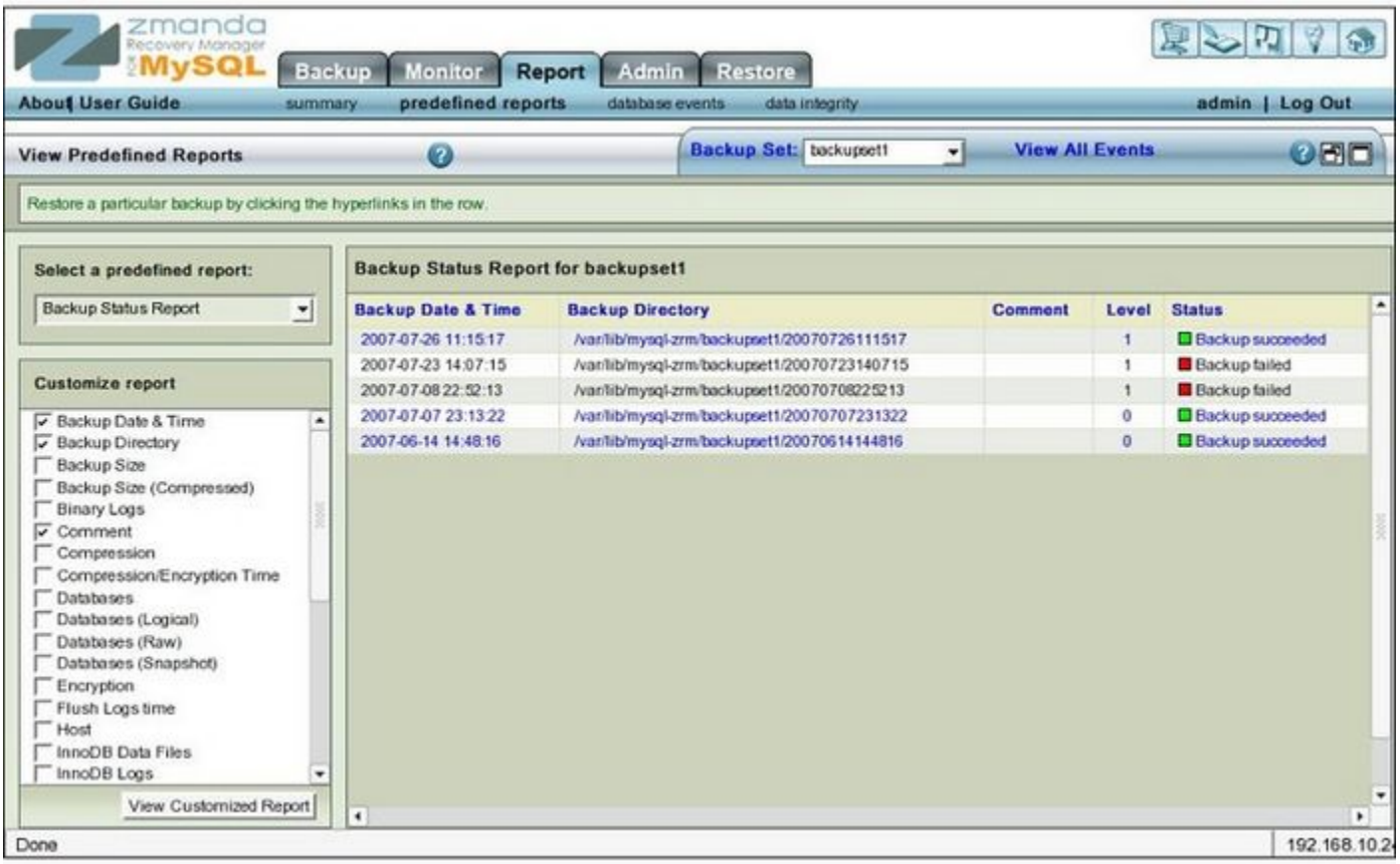
The **Backup Report** has columns that display **Level, Database and Tables**, corresponding to the **Backup What** parameters. It also has a column that shows the **status** of the run.

Application Impact Report



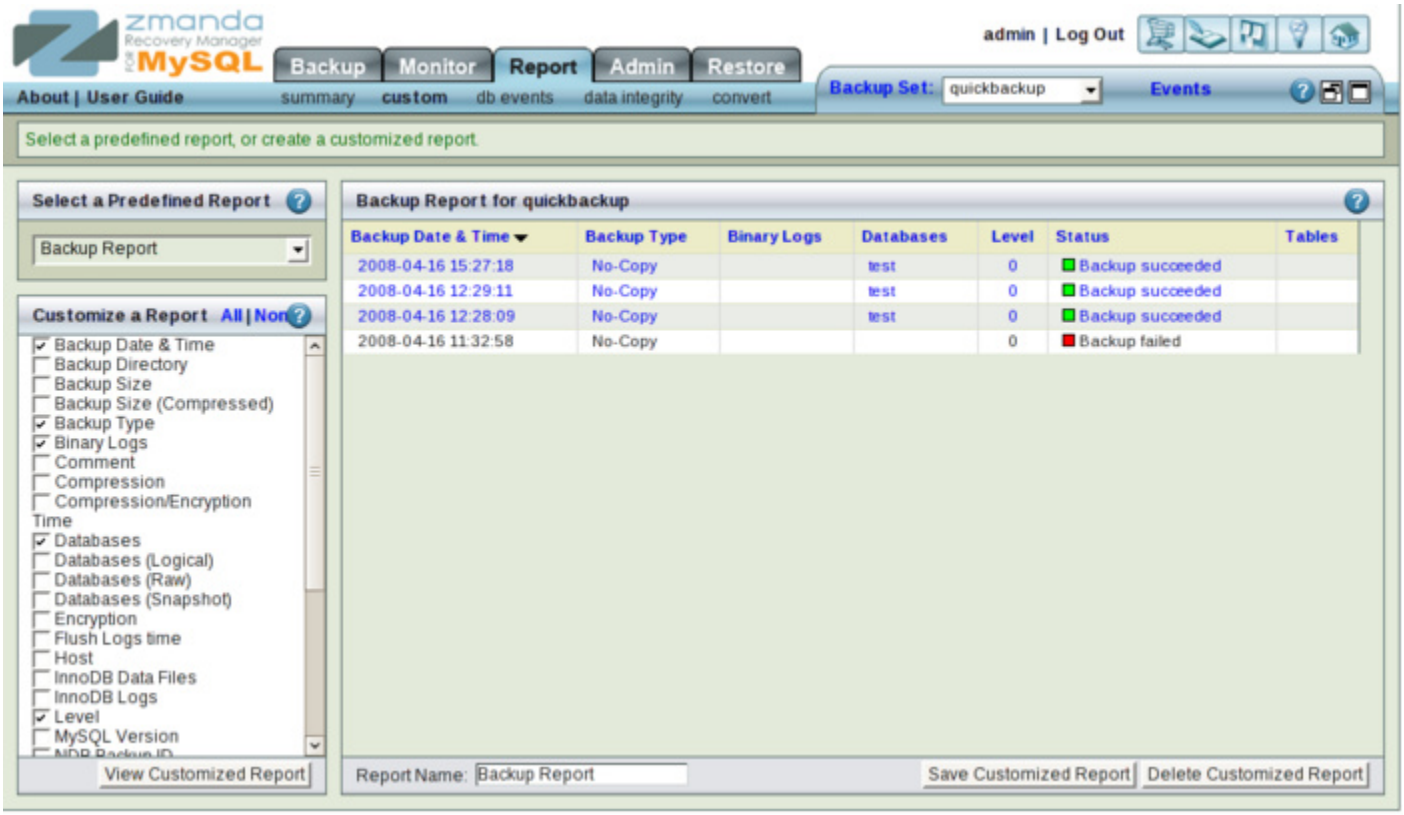
The **Application Impact Report** displays the **Read Lock Time** and **Total Time** for each run. These allow you to determine how the backup run affected database application performance.

Backup Status Report



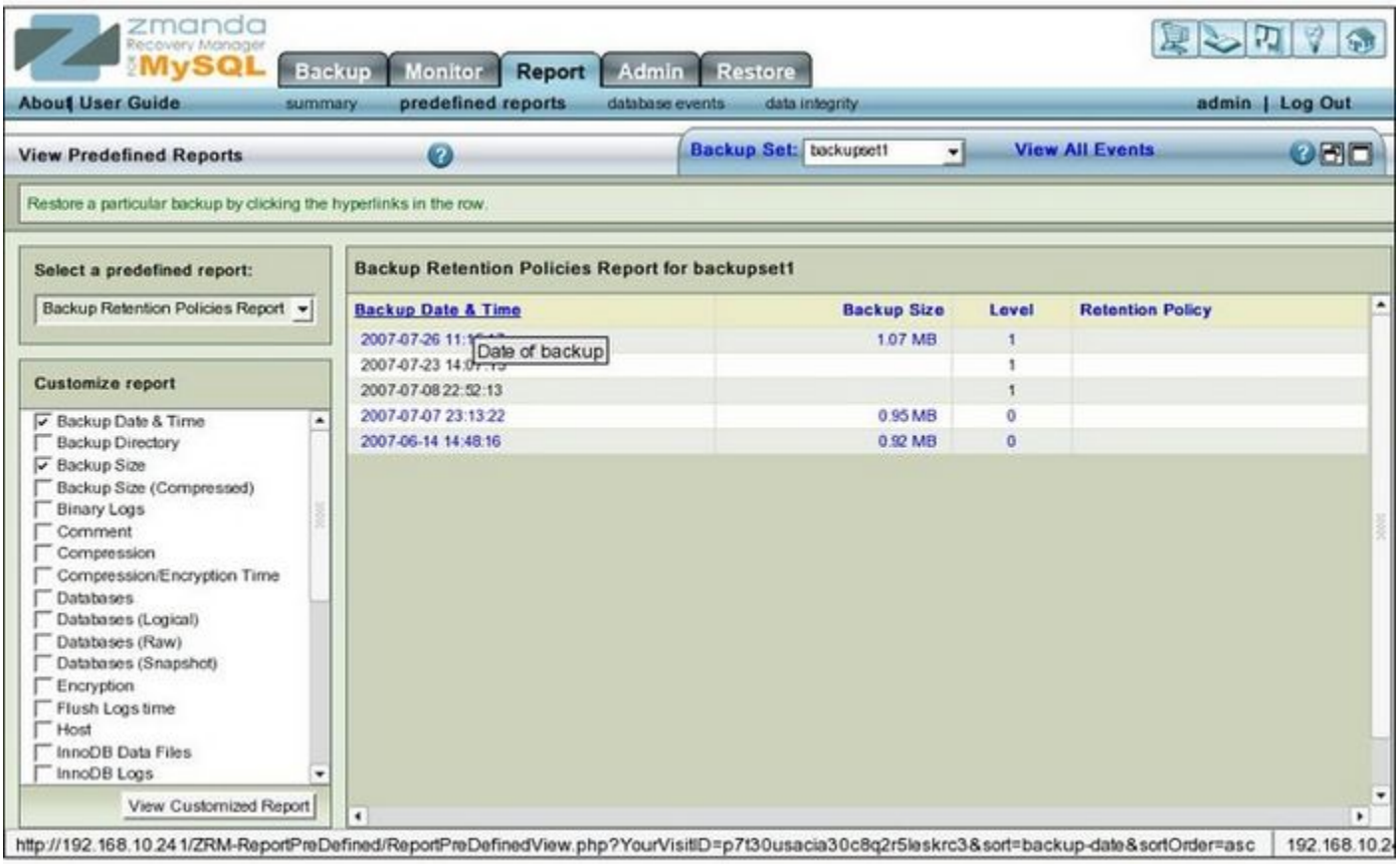
The **Backup Status Report** displays the **Status** and the **Destination Directory** where the data has been backed up. The **Status** of the run is also displayed.

Backup Method Report



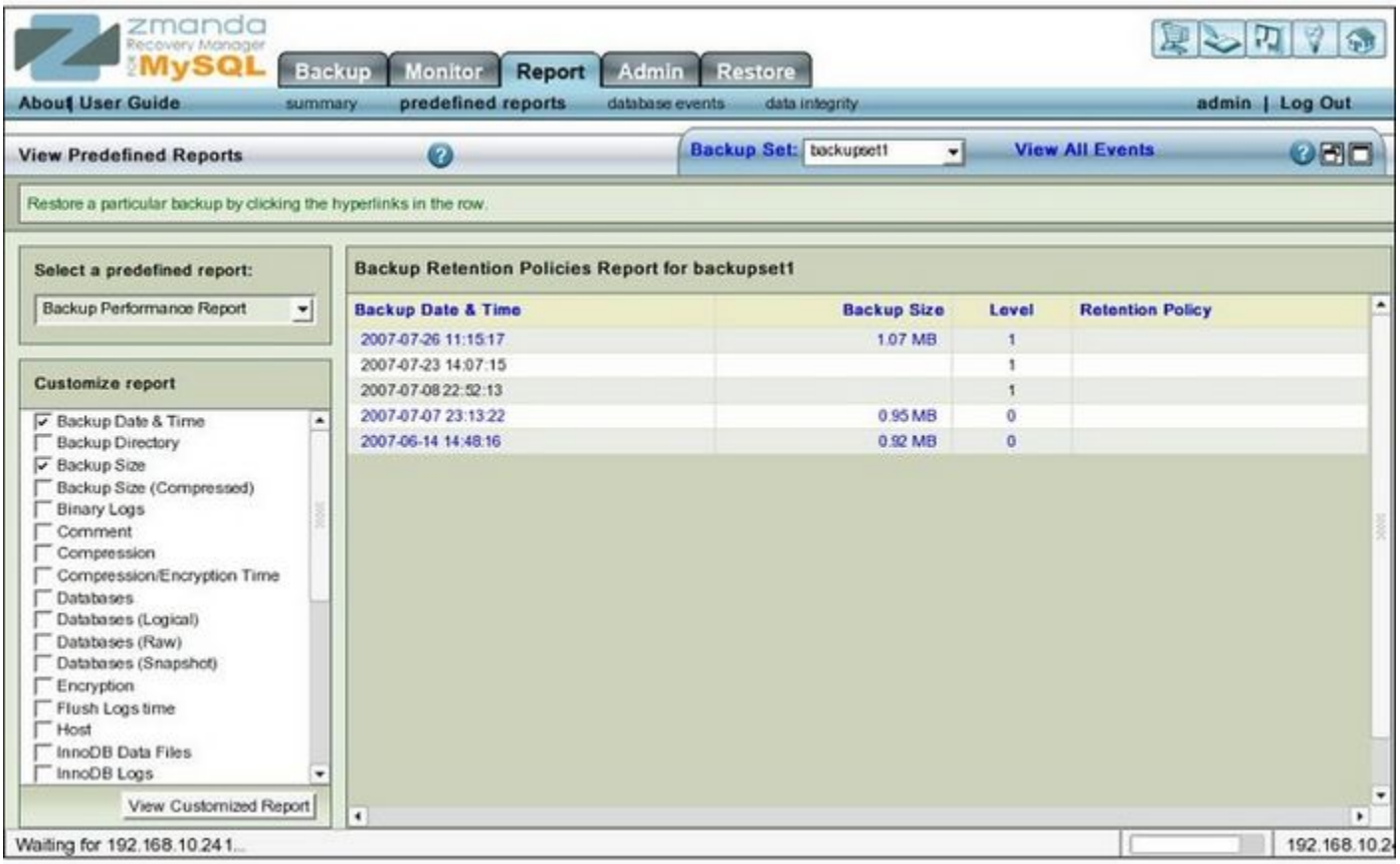
The **Backup Method Report** displays the backup method (i.e. **Databases Logical, Databases Raw and Snapshots**) used to create the backup. The data in these columns will not change across runs. By changing the backup set selection in the drop down box at the top of the page, can quickly see which backup sets have what methods.

Backup Retention Policy Report



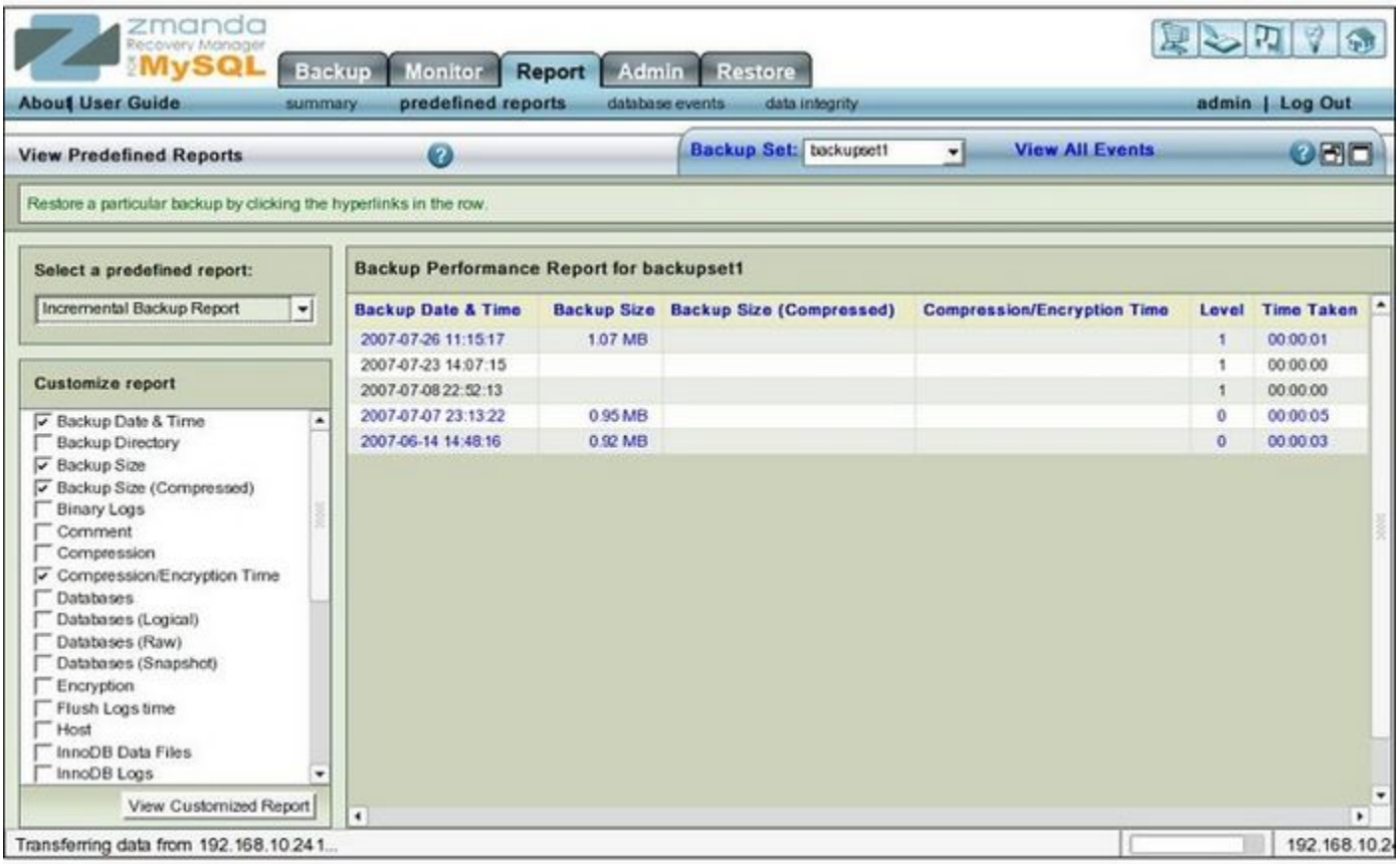
The **Backup Retention Report** shows the retention policy for the backup set. A blank column indicates that no purge policy has been set for the backup set.

Backup Performance Report



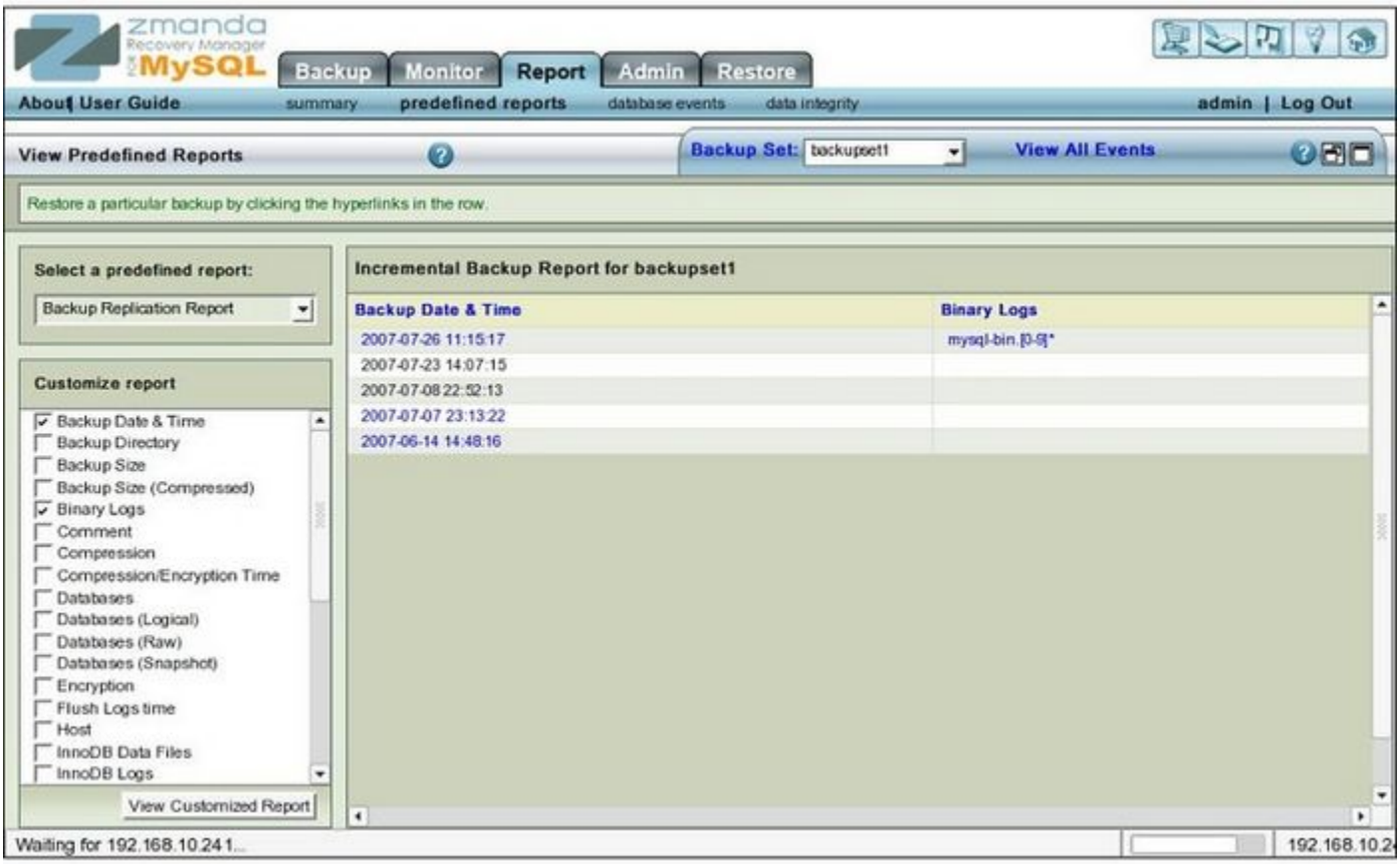
The **Backup Performance Report** displays compression and performance statistics for the backup set.

Incremental Backup Report



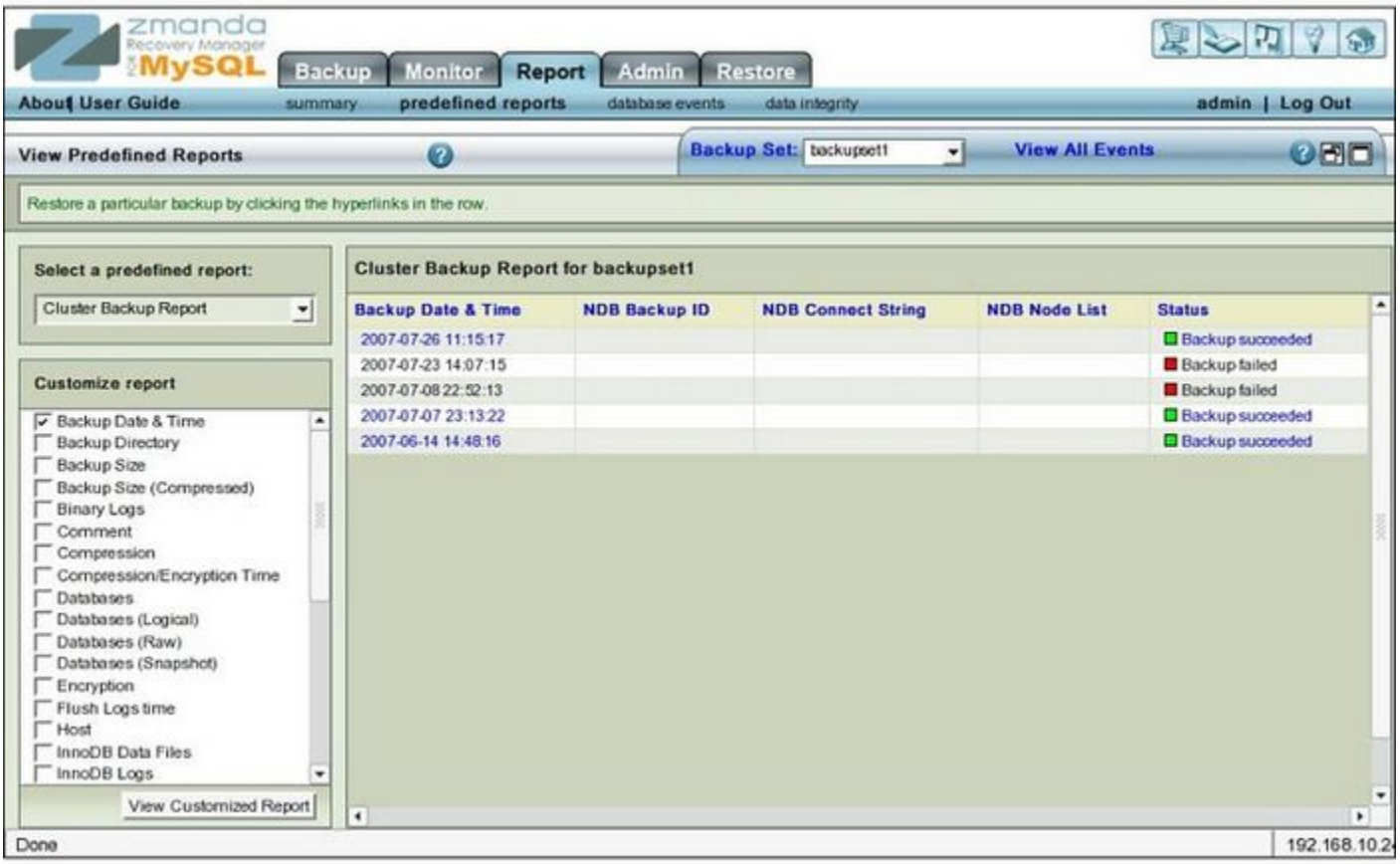
The **Incremental Backup Report** shows the kind of backup performed: Incremental Backup Sets will display a value, while full backups display a blank column.

Replication Backup Report



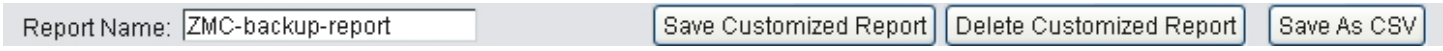
The **Replication Backup Report** has two columns. **'Replication Files'** and **'Slave Load Files'** show values when Replication has been set for the backup set.

Cluster Backup Report



The **Cluster Backup Report** has three columns. **'NDB Backup ID' , 'NDB Connect String' and 'NDB Node List'** that will show the same values until changed by the User. The **Status** column allows users to judge the impact of changing these parameters.

Customized Report



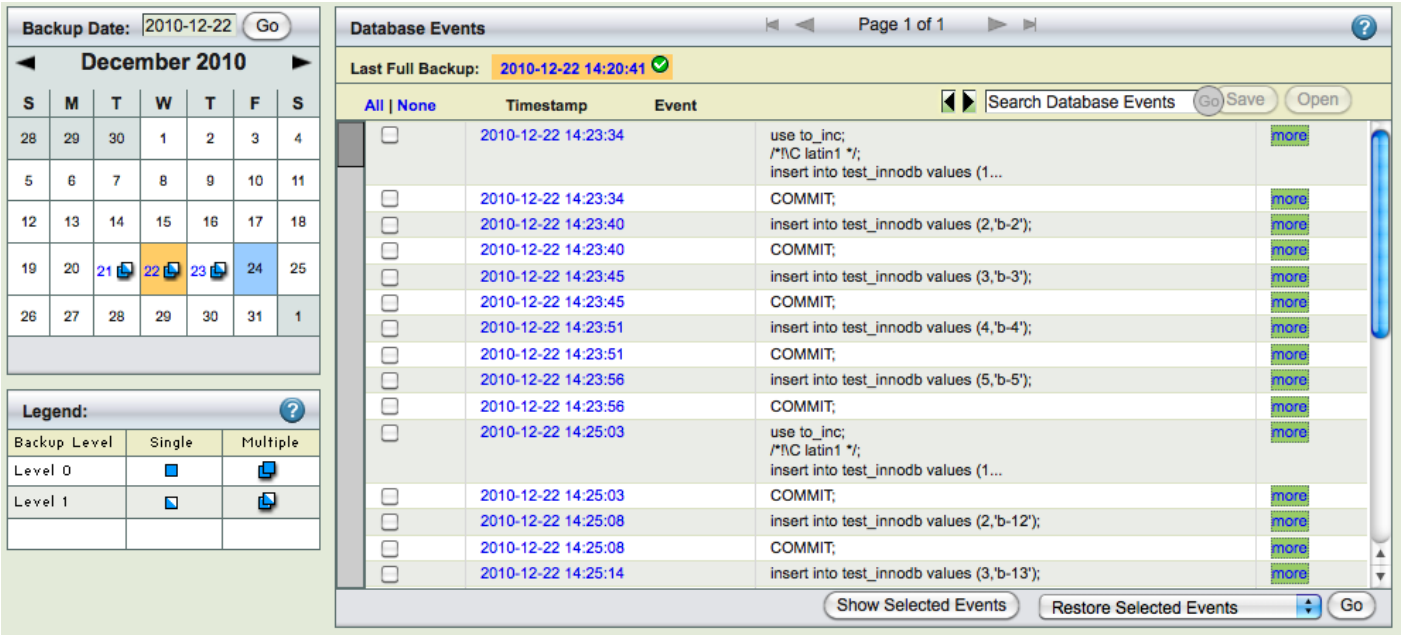
You can name the customized report so that it can retrieved later. All ZMC users can retrieve the report. The report name appears as part of list of Predefined reports.

Save as CSV button allows users to save custom reports in CSV format. This format can be read by any spreadsheet or report analysis tool. Before saving a custom report as CSV, you must name the report using **Save Customized Report**. The CSV output is saved in a file named *zrm3_1-<custom report name>.csv*

Database events report

The **Database Events** page lets you find and view all the database events logged and copied during incremental backups. This is useful when you need to find a particular malicious or erroneous transaction so you can roll back the unintended change. Once you find the problem event, you can easily launch a restore operation that will roll back all changes to the moment before the damage occurred. To back up and restore at the transaction level requires that binary logging is enabled on the MySQL server.

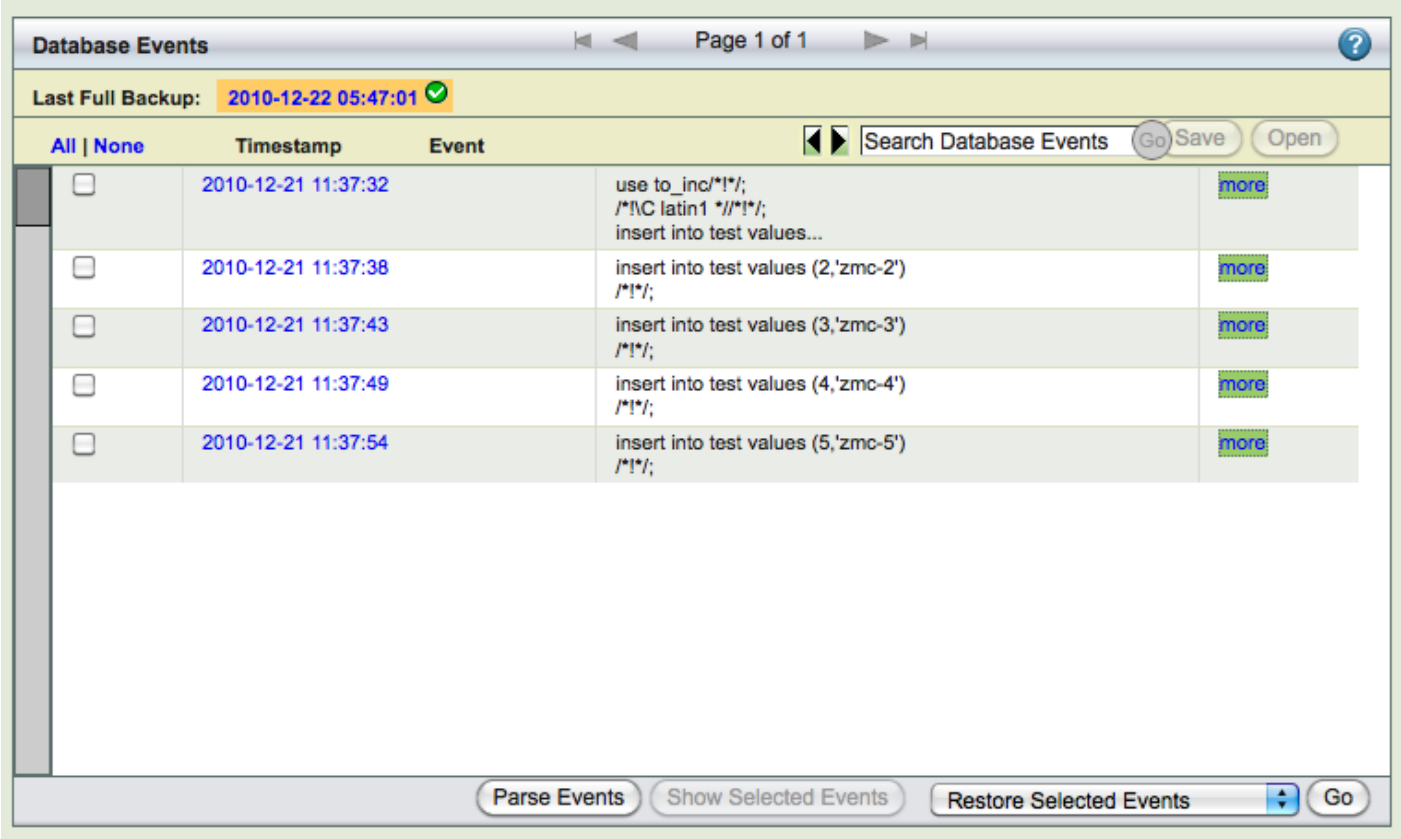
It is possible to restore selected events or undo selective events from the database. You can also search for errors and save the search query for future searches. This will make identification of erroneous events easier.



The left panel displays a calender control that lets you select a backup icon from dates on which single or multiple backups where processed.

The right panel features controls that help you locate events within the selected backup. When you open the page, the right panel displays events from the selected backup date for the selected backup set. Binary log backups are stored as part of full and log incremental backups.

Parsing of binary logs from the backup images can take time (up to few minutes). To refresh the report page using **Parse Events** button at the bottom of the page (as shown below) The backup images are parsed again and report is re-populated.



Selecting Backup Date

Calender Control Panel

There two methods for selecting a backup:

- You can click a backup icon on the calendar.
- You can enter the backup date (*mm*/*dd*/*yy*[*yy*]) at the top of the left panel and click **Go** to select a backup date.

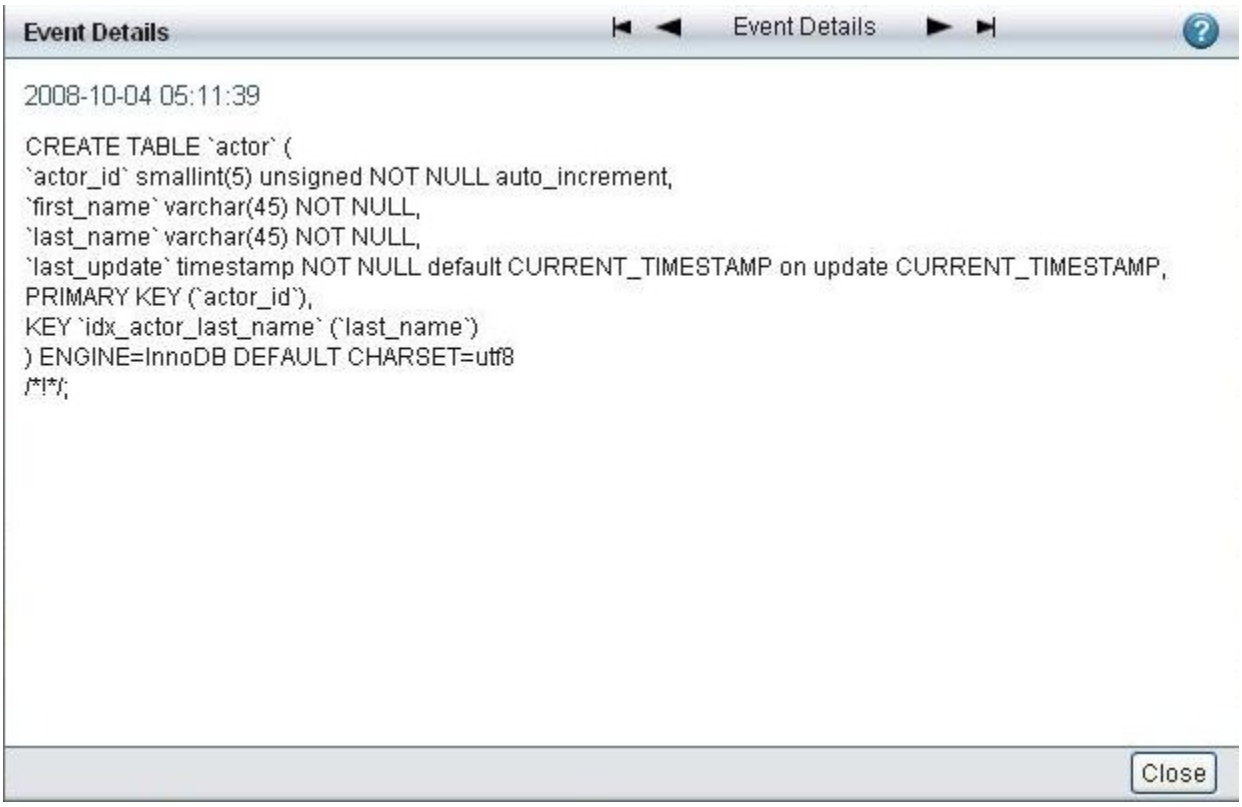
The calendar shows the dates on which full backup (level 0) and incremental backups (level 1) were performed. If the multiple backup runs happened on a calendar day, it is shown with a different icon. You can select any date that has full or incremental backups.

The binary logs in the backups are parsed and displayed in the database events window. When user selects a backup date, the backups done on that date are parsed. You can use **Parse Events** button to parse the backup image again. Page refresh will not parse backups again.

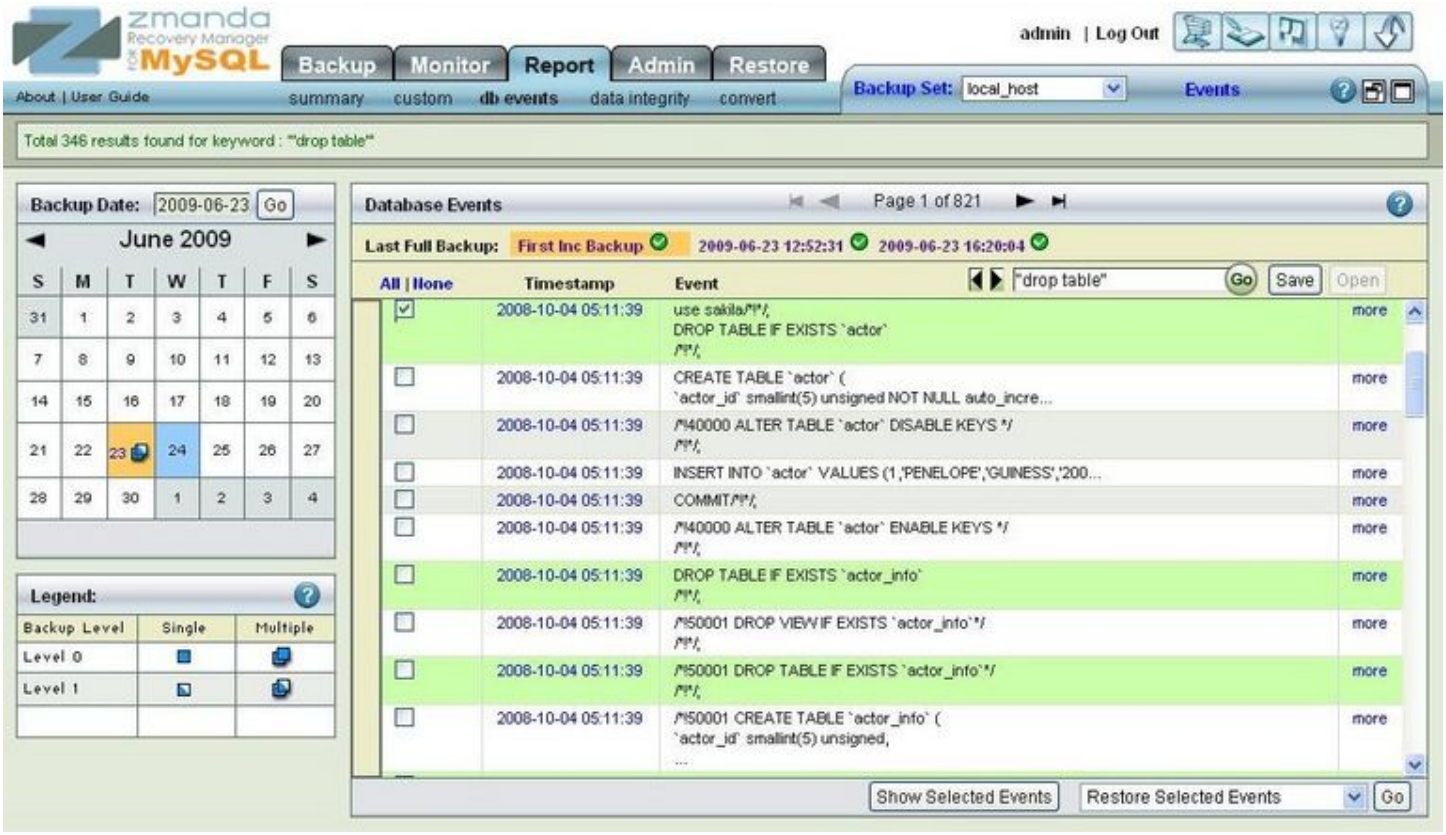
Note: If no backup exists for the selected date, then no data is displayed. This does not necessarily mean that database event logs do not exist for that date; they may be available from a binary log that was backed up subsequently. In other words, the Database Events log includes all data from a given backup to the next backup that included logged events.

Database Event Viewer

After you have selected a backup that includes log data, the events viewer lists the events it contains in a summary list. You can click on the **more** link to get complete description of the event as shown below.



Database Event Search

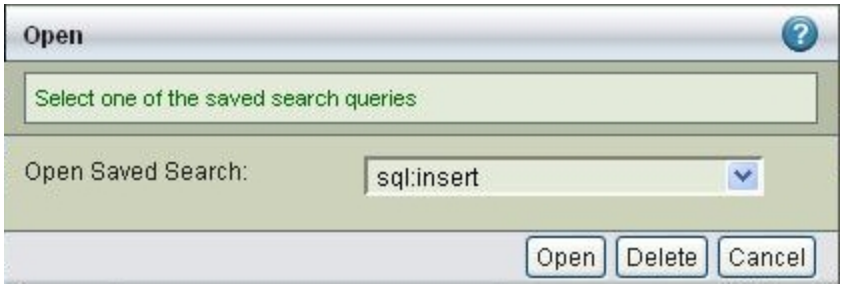


You can enter a text string to search for specific database events. You can use MySQL fulltext search Boolean operators to further control the search. For example, you can quote entire phrases to find them, and use plus (+) operator to refine searches on individual words.

- + A leading plus sign indicates that this word must be present in each row that is returned.
- - A leading minus sign indicates that this word must not be present in any of the rows that are returned. You can use both + and - together
- * The asterisk serves as the truncation (or wildcard) operator. Unlike theother operators, it should be appended to the word to be affected. Words match if they begin with the word preceding the * operator.
- " A phrase that is enclosed within double quote (“"”) characters matchesonly rows that contain the phrase literally, as it was typed. Search "open source backup" we will do the search for exact that phrase only.

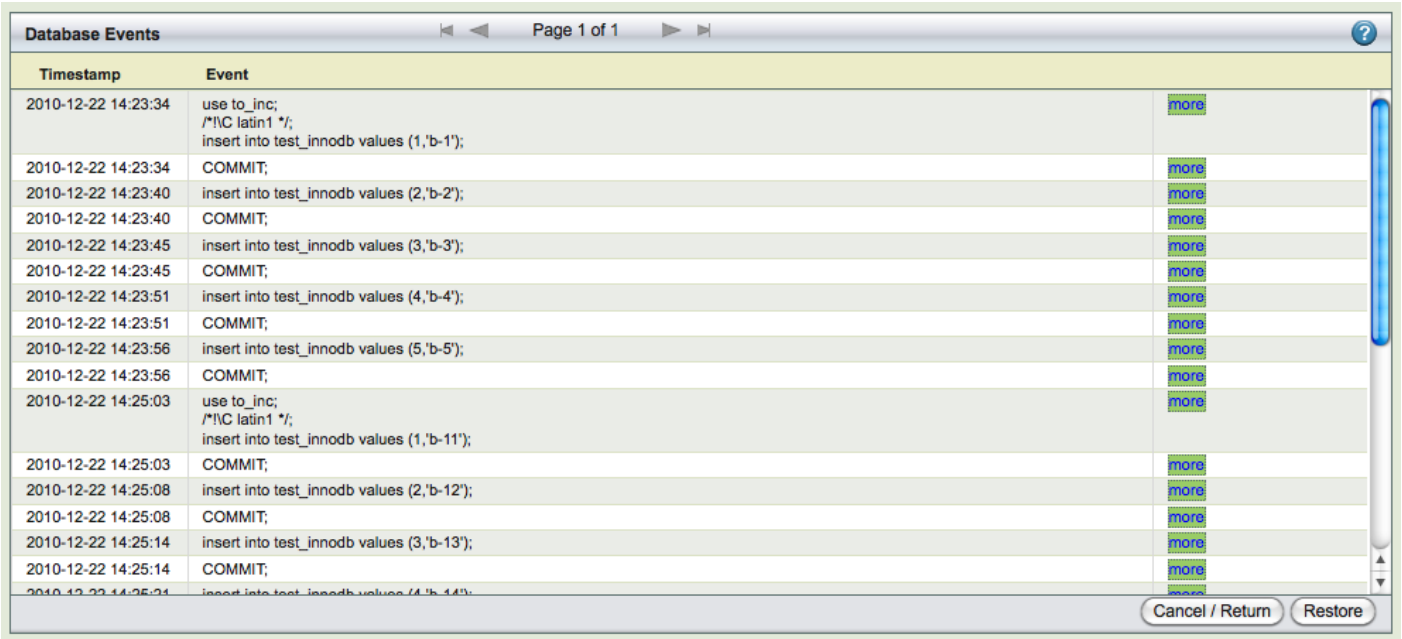
All the search results are selected. The search results are also highlighted. It is possible to use the left and right arrow buttons next to the search box to go from one search result to another.

Search queries can be named and saved for future searches (Use *Save* button next to the search box). Clicking *Open* button next to the search box to use saved search queries. Saving search queries can help in performing the same queries on multiple backup sets or multiple backup dates. Following figure shows the dialog that is displayed when *Open* button is clicked. Users can open a saved search for searching and delete a saved search in this dialog window.



Launching Restores from the Database Events Reports

After selecting the event(s) using the checkbox next to each event, you can restore to a selected event (of course, only one event should be selected) or restore only the selected events or restore everything except the selected event (undo selected events). Multiple events can be selected for restoration or undo the effects of the selected events. Restoring to a selected event is equivalent of point-in-time recovery i.e, all database events starting from last full backup to the specified event are restored.



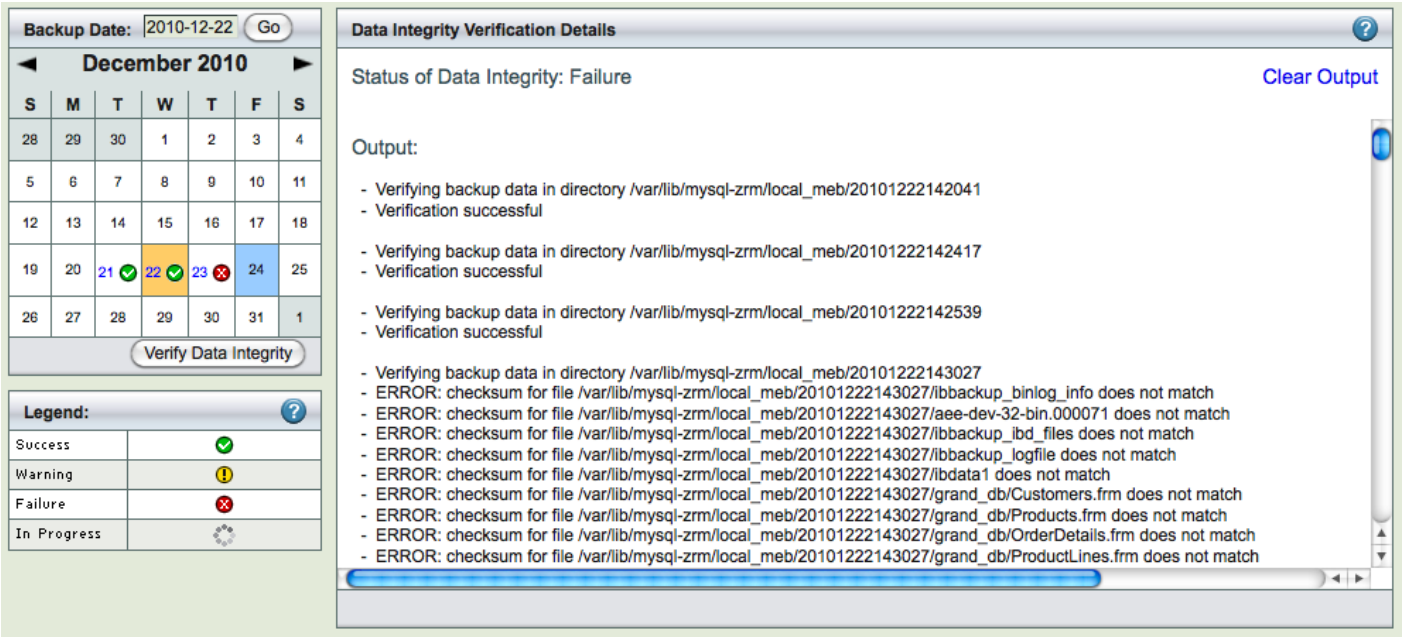
Show selected events button shows all the selected events in one page as shown above. This can be used to review selected events and confirm the restoration process. Clicking **Go** in the Database event report or **Restore** in the selected events page takes you to **Restore What** page to continue with restoration process.

Selective restores should be performed carefully. Otherwise, there can be restoration failures or data loss. Few important points that user must be aware of:

1. ZRM does not have capability of verifying if the selected events for restoration or the *undo* operation makes sense from the database or the database application. It is up to the user confirm that the restoration of events will keep the database consistent from application point of view and there will be loss of data.
2. While performing selective restore, database constraints should be met after selective restoration. Selective restore can fail due to database constraints.
3. The mysql user used for restoration (defined in **Backup What** page) should have privileges to perform the selected events being restored. The privileges should be granted to the user before attempting restoration.
4. Selective restoration depends on the time to be in sync between the ZRM server and MySQL server in case of remote MySQL servers. Otherwise, correct events may not be restored.

Backup image integrity report

The Data Integrity page provides a tool to query and confirm that the backed up data has not been altered since it was backed up. Verifying the integrety of data is important when you move a backup image from the hard disk to save space, and then move it back online in preparation for restoring it. In all circumstances, verifying the backup image is recommended before restoring databases from it.


Backup Date: 2010-12-22 Go
December 2010
S M T W T F S
28 29 30 1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31 1
Verify Data Integrity
Legend:
Success
Warning
Failure
In Progress
Data Integrity Verification Details
Status of Data Integrity: Failure
Clear Output
Output:
- Verifying backup data in directory /var/lib/mysql-zrm/local_meb/20101222142041
- Verification successful
- Verifying backup data in directory /var/lib/mysql-zrm/local_meb/20101222142417
- Verification successful
- Verifying backup data in directory /var/lib/mysql-zrm/local_meb/20101222142539
- Verification successful
- Verifying backup data in directory /var/lib/mysql-zrm/local_meb/20101222143027
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/ibbackup_binlog_info does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/aee-dev-32-bin.000071 does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/ibbackup_ibd_files does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/ibbackup_logfile does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/ibdata1 does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/grand_db/Customers.frm does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/grand_db/Products.frm does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/grand_db/OrderDetails.frm does not match
- ERROR: checksum for file /var/lib/mysql-zrm/local_meb/20101222143027/grand_db/ProductLines.frm does not match

- The left panel contains a Calender Control that allows Users to navigate to the Backup Date when the backup took place.
- The right panel displays status and results of the verification process.

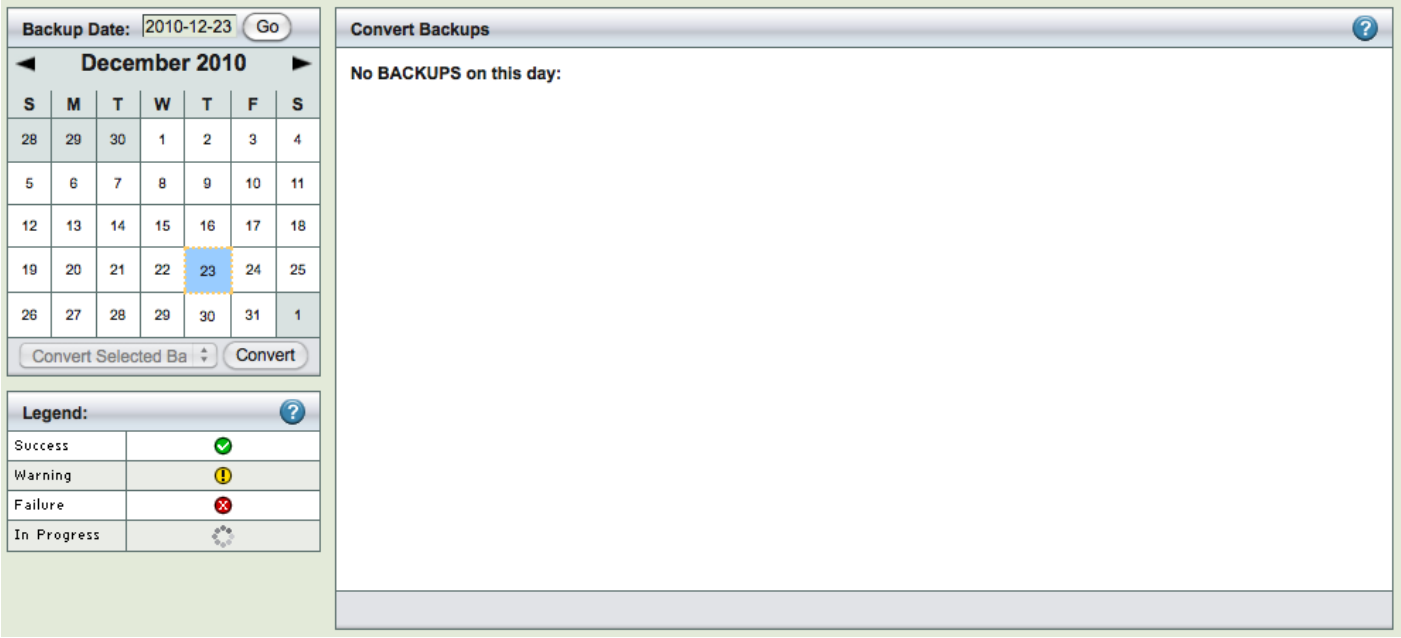
Clicking the **Verify Data Integrity** button starts the verification process. After you click the **Verify Data Integrity** button, the right panel shows the progress and result of the verification.

When you open the page, the Status panel displays a message 'No Task launched' until you click the **Verify Data Integrity** button. The message changes to **Running** after you click **Verify Data Integrity.** The verification process assumes that the backed up data is present in the same directory where it was originally backed up. If the data is not present, the verification fails.

Note that Quick Snapshot backups involve no data transfer, and therefore cannot be verified.

Convert Quick backups to Standard backups

This page displays any Quick (No-copy) backups that have been performed, allowing you to select and convert the quick snapshot to a standard ZRM backup stored on the ZRM server:



Quick snapshots and the reasons for converting them are described in Snapshots overview section. The **Convert** page is divided into two panels:

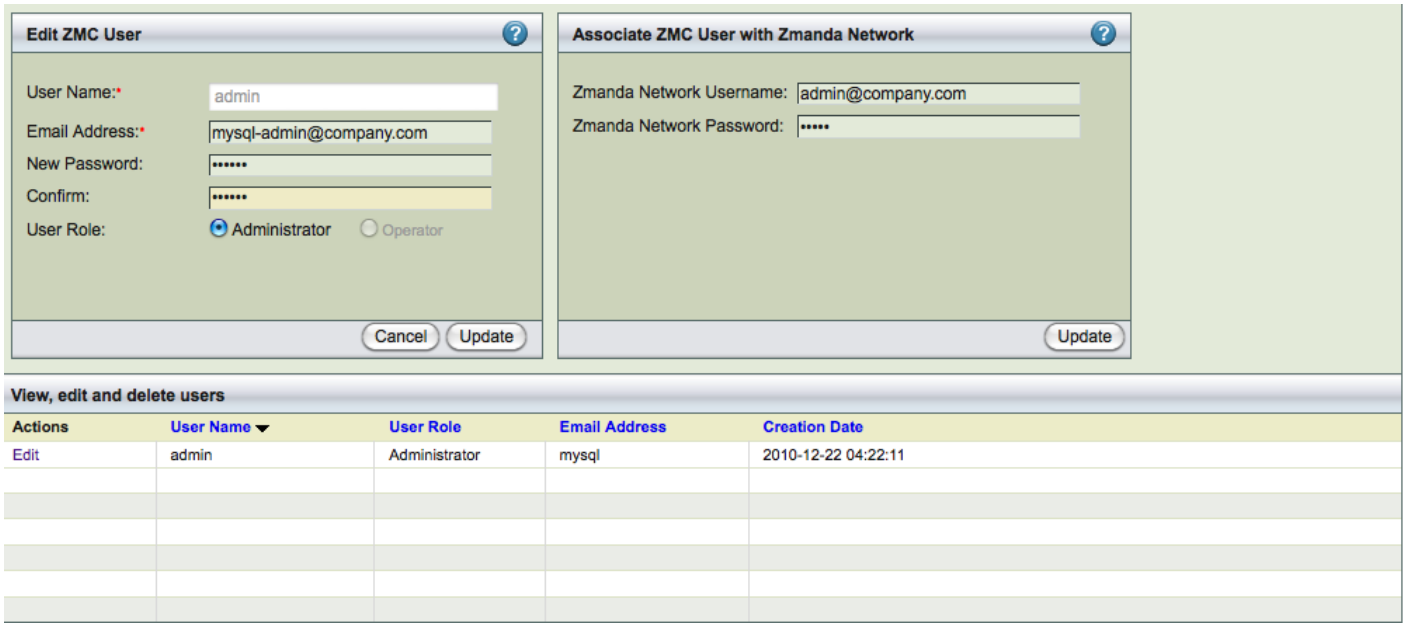
- The left panel contains a Calender Control that let you select a backup by date and time to convert. Backups are displayed in the calendar as indicated by the legend. Quick full backups are only displayed.
- The right panel displays the results of the verification process.

You can either enter the date (*mm*/*dd*/*yy*[*yy*]) and click **Convert**, or click on backup icon in the calendar. When you click **Convert**, ZRM for MySQL begins converting the backup for that date.

Clicking the **Convert** button starts the conversion. Status is displayed as described below. Note that once a quick snapshot has been successfully converted, it is no longer available for selection in the calendar.

ZMC User administration

The **Admin Users** provides a convenient way to create, edit, view and delete ZRM for MySQL users. It also lets you register ZMC users with the Zmanda Network. Please take the time to register users with the Zmanda Network; registration is required for accessing the Zmanda Network Knowledgebase from within the Zmanda Management Console.



ZMC User

The default username is **Admin**, with a password of **admin**. You should change this upon first login. ZMC users are not linked to LDAP or Active Directory services.

User name

Choose a user name. The name can be any alphanumeric string. Dots (.) and dashes (-) are allowed; spaces are not.

Email address

This address will be used if the user's password must be recovered (see the Login page).It is important to configure the ZRM server to send emails. Otherwise, emails sent to this address will not be delivered.

Password

The ZMC user password. You must enter it twice for confirmation purposes.

User Role

ZRM for MySQL implements a Simple RBAC (Role Based Access Control) scheme: A user can either be an **operator** or an **administrator.**

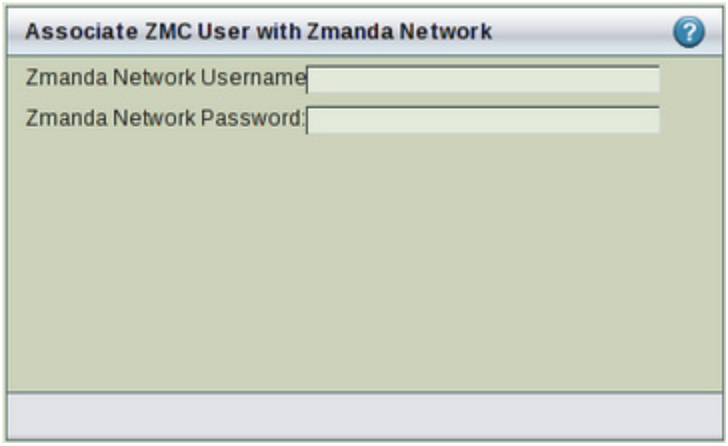
An administrator can:

- create backup sets and assign their ownership to anyone in the system.
- edit or delete backup sets owned by any other user in the system.
- see all events and all alerts.

An operator can:

- Create backup sets owned by that operator.
- Edit or delete backup sets owned by that operator.
- See all alerts, but only the events attached to the backup set owned by that operator.

Associating with Zmanda Network



Zmanda Network username and password

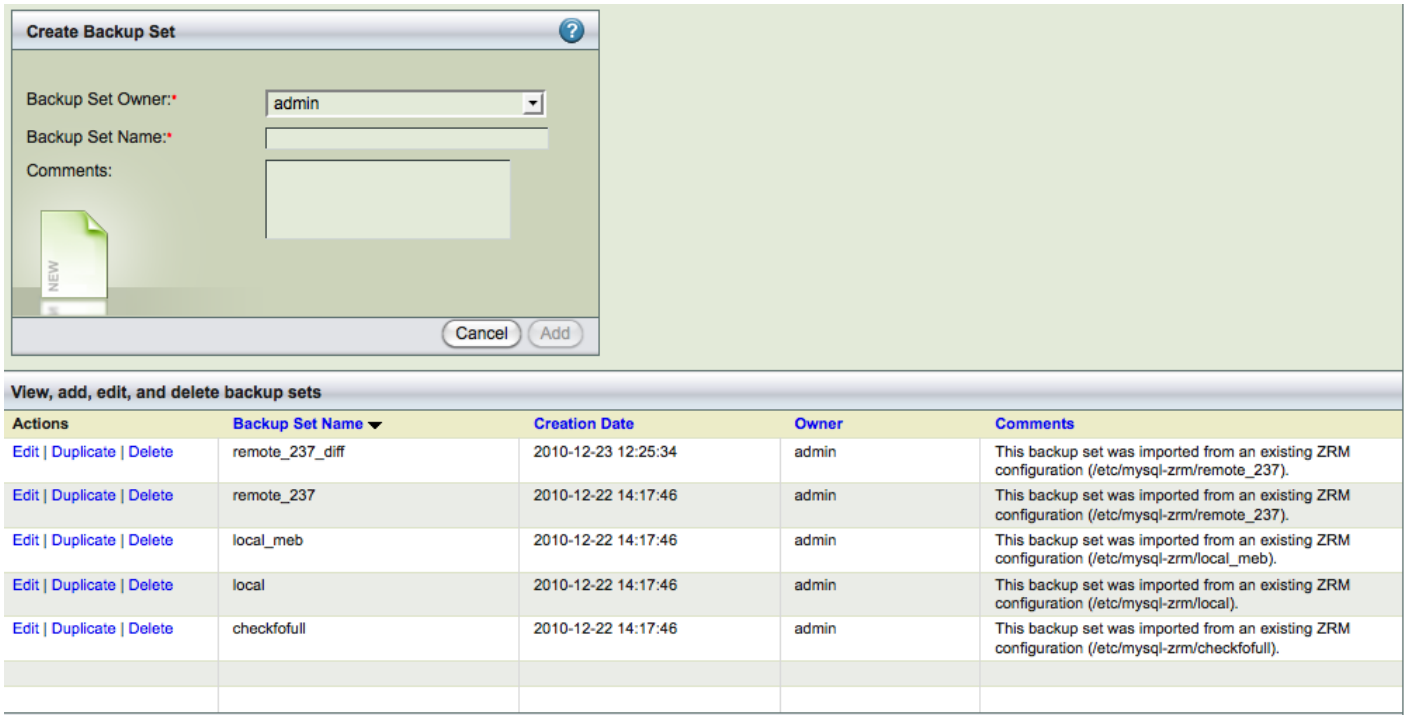
These fields lets you associate the ZMC user with an existing Zmanda Network login ID. Enter the username (i.e., the email address) and password that were provided during Zmanda network registration. Click the **Add** button to validate the Zmanda network login details. If the login credentials are correct (i.e., they match those in the Zmanda Network), the change is updated on the ZMC user list.

Editing/Deleting ZMC User

To edit or delete an existing user, click either the **Edit** or **Delete** link next to that user. When you click **Edit**, the contents of the row are displayed in the fields above for editing.
All Users can edit their own password and email address; only users with administrator privileges can edit or delete another user's account.

Managing Backup Sets

The main functionality of the **Admin Backup Sets** page is to create, edit, delete backup sets. It also provides a way to duplicate backup set configurations.



- The top panel, **Create Backup Set and Comments**, lets you create backup sets along with comments.
- The bottom panel, **View, duplicate, edit and delete backup sets**, lets you manage backup sets.

Backup Set Creation

Backup Set Owner

Specify which ZMC user owns the backup set. By default, the owner is the user who created the backup set. All users can transfer their ownership to another user by editing the backup set and clicking the **Update** button.

Backup Set Name
Specify an unique and descriptive name for the backup set. The name can include any alphanumeric characters, along with periods and dashes (as long as they are surrounded by alphanumerics; for example: **admin-backup** is allowed, whereas **admin--backup** is not). Spaces are not allowed.

Comments
Enter an optional comment that describes the purpose of the backup set.

Click Save to create the backup set. After it has been saved, it will be added to the list of backup sets. Other than name and comments, it will inherit all of its settings and options from the Site Settings page. Note that if the backup set list consists of multiple pages, you will have to page to the end of the list to see the new backup set.

Modifying Backup Sets

The Duplicate, Edit and Delete links let you manage the backup set list. Click on the appropriate link to the left of the backup set to perform the operation.

Duplicate a backup Set
Adds a duplicate copy of the given backup set to the bottom of the list. This allows you to create a backup set that inherits its properties from the source backup set rather than the **Set Site Defaults** page. Duplicate sets are named after the original, with the **_copy** suffix appended. Edit the backup set as necessary to meet your requirements.

Edit Backup Set
Click to edit the given backup set.

Delete Backup Set
Removes the given backup set (and all its settings) from the ZMC. The ZMC will prompt for confirmation; proceed with caution as there is no way to undo the deletion. If you attempt to delete a backup set that is active (i.e., selected from the **Backup Set** dropdown at the top right of the page), the ZMC will take you to the **Create a New Backup Set** page after confirmation, otherwise you will remain on the **Admin Backup Sets** page. Note that deleting a backup set has no affect on backups already completed using that set or the backup images.

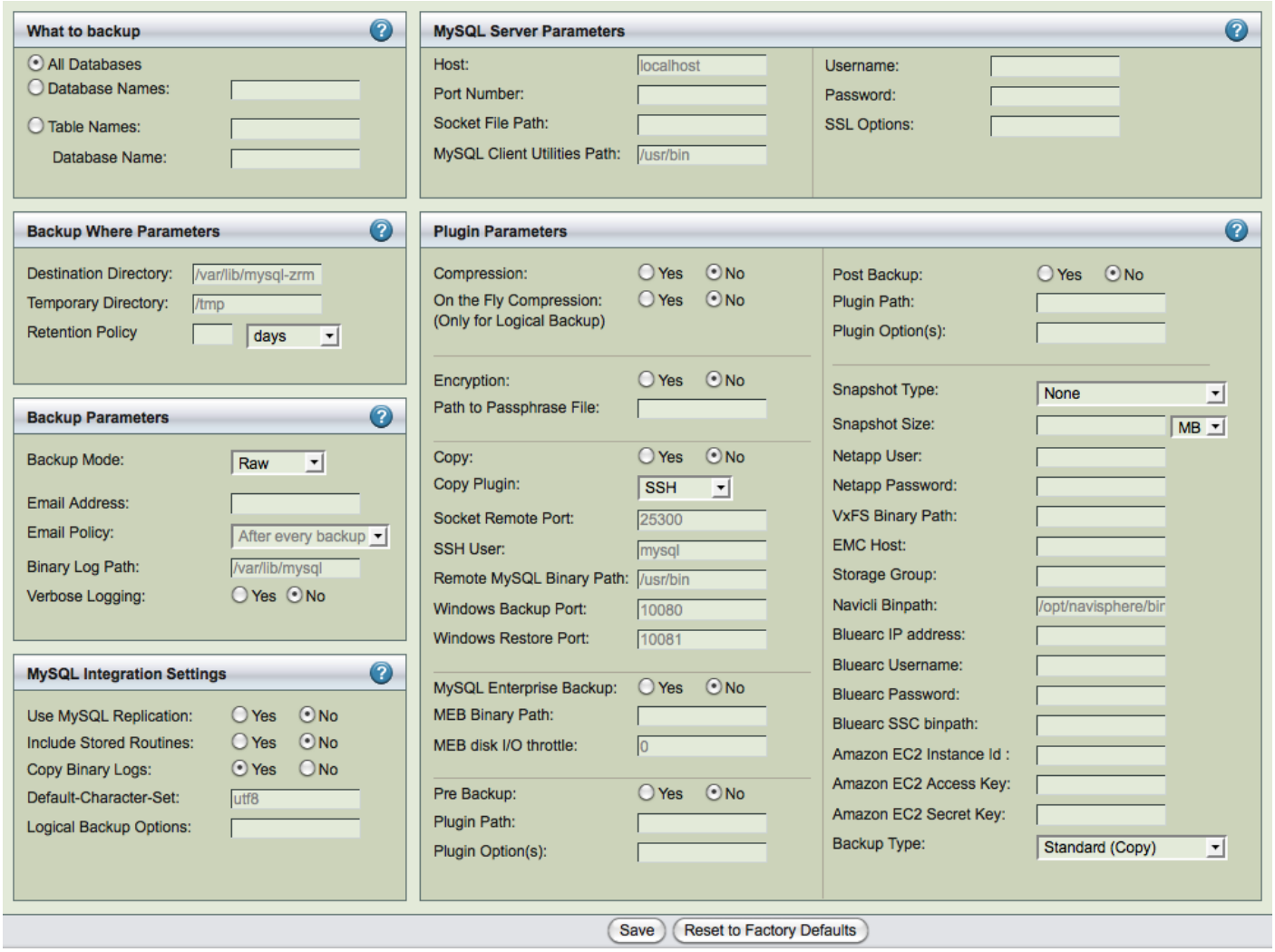
Set Site Defaults

The **Set Site Defaults** specifies the global default values for backup sets. Setting the appropriate defaults for your site will make backup set configuration much easier. Changes set here take effect when the next backup run is executed.

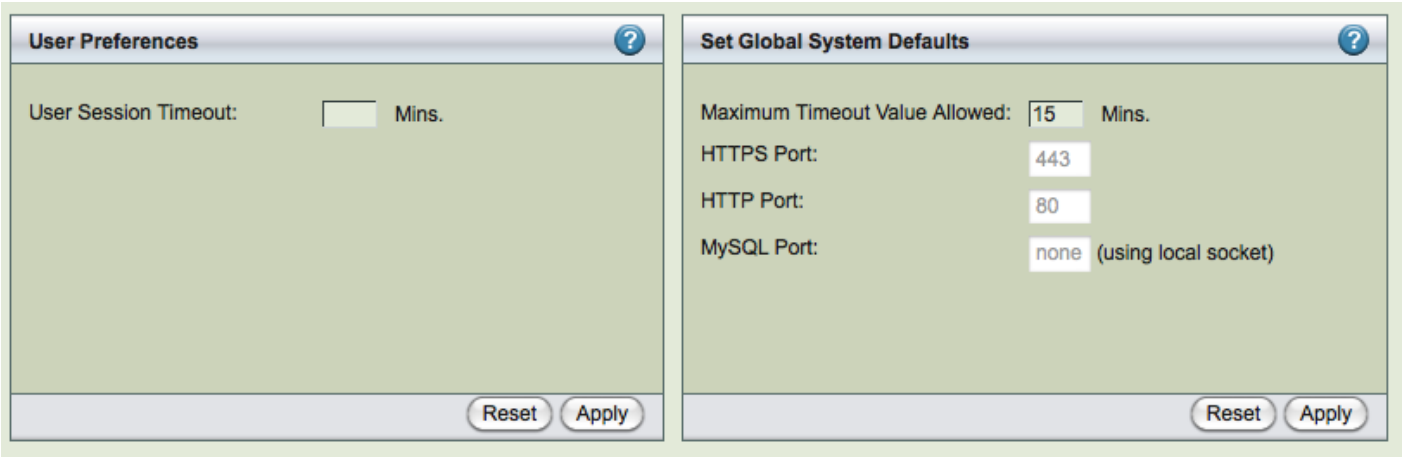
If you have multiple MySQL servers backed up by ZRM, you should store all parameters common to all MySQL servers as site defaults. These values are inherited by all backup sets. You can configure MySQL server specific information such as IP address of MySQL server in the backup set configuration.

Backup Summary provides information about the configuration parameter value and where it was set - **Factory Settings** (ZRM default value), **Site Defaults** (Set in **Set Site Defaults** page) and Backup Set (Set in Backup What/Where/When/How pages).

The values can be modified by **Administrator** user.



Admin Preferences



The **Admin Preferences** page lets you set the user and session timeouts and look at the ports used by ZMC Apache processes and MySQL processes.

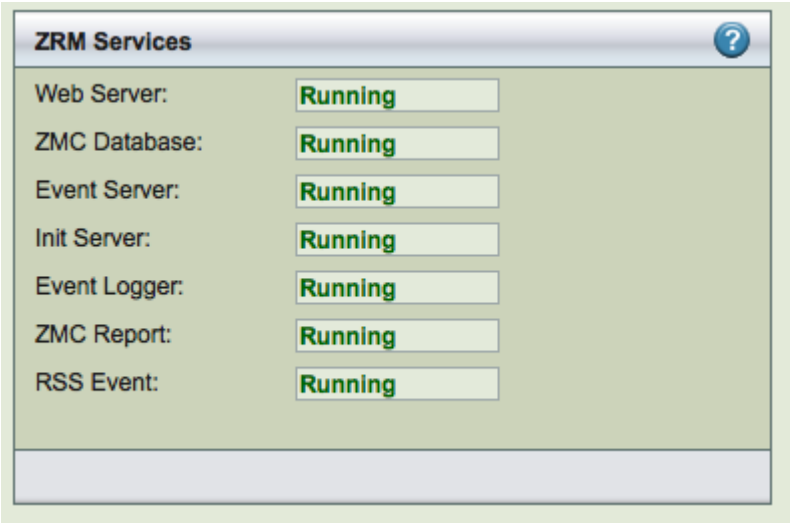
User session timeout
Set the number of minutes you (i.e. the user who is logged in to ZMC) is allowed to stay logged in with no activity. The default is **0**, which allows the user to stay logged in the maximum number of minutes allowed by the **Session Timeout** setting (described below). The user session timeout cannot exceed the session timeout.

Maximum Timeout Allowed
Set the number of minutes any user is allowed to stay logged into the ZMC. The default is 15 minutes, with a maximum of 300 minutes allowed. The setting takes effect upon the next login. A long session timeout value (for example, over an hour) can have security implications, so excercise caution in setting the value too high.

HTTP, HTTPS port
Ports used by Apache web server. This is a display field. To change the value, please see the Zmanda Network Knowledge Base how to article.

MySQL port
ZMC MySQL services do not accept requests from TCP/IP port. All communication is done using local socket file.

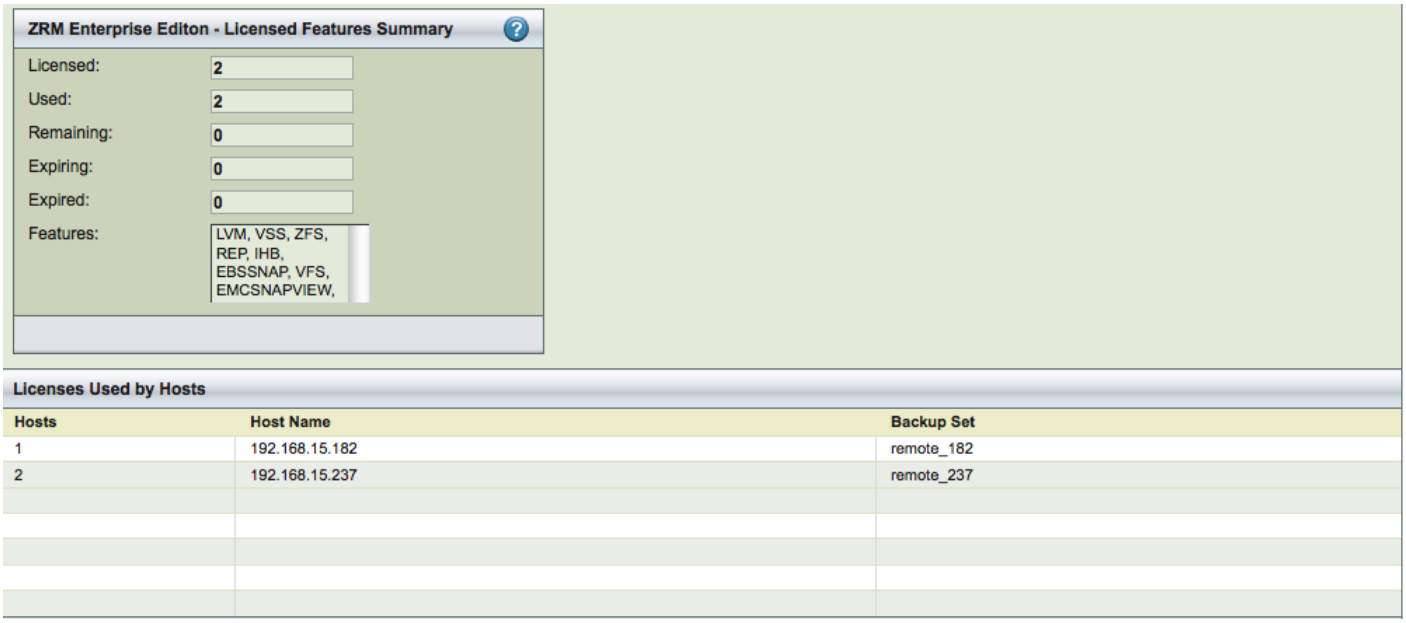
ZMC services status



The *Admin Services* page shows the status of all ZMC processes. There are multiple components of Zmanda Management Console. All components have to be in *running* state for Zmanda Management Console to function correctly. The "running" state does not imply that they are consuming CPU/Memory resources. It implies they are waiting for requests.

ZRM License status

ZRM Enterprise product is a licensed product. The license file can be downloaded from Zmanda Network. The license file should be stored in the ZRM server. This page (shown below) shows the licensed quantities, subscription time and how they are used.



The **Licensed Features Summary** panel shows the features that are licensed. All features and MySQL servers Hosts are licensed for the subscription period.

Licensed

The number of MySQL servers (IP addresses) that can be backed up by this ZRM server that have valid subscription.

Used

The number of MySQL server licenses used by backup sets

Remaining

The number of MySQL server licenses ununsed.

Expiring

The number of MySQL server licenses expiring in next 30 days

Expired

The number of MySQL server licenses that have expired

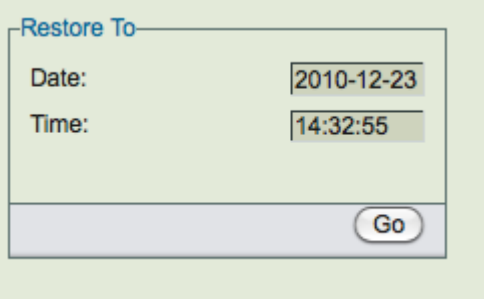
Features

The list of licensed snapshot features

Licenses Used by Hosts table shows the list of MySQL server Hosts, their IP addresses and the backup sets that configured for the MySQL server.

Restore What

This is the first step in the restoration process.



Restore To

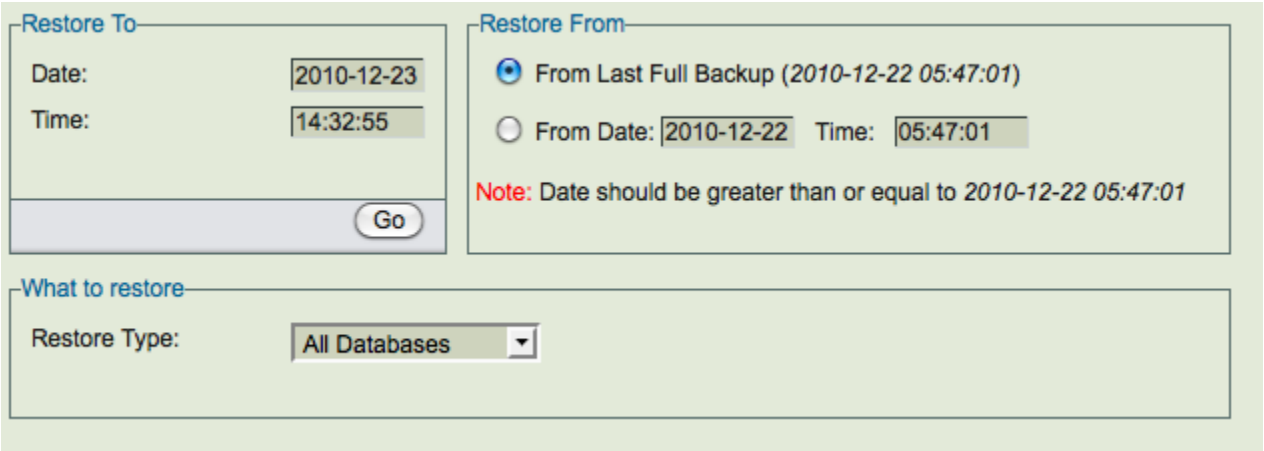
If you came to this page using one of the report links (which is recommended), the date and time will be filled in based on the link that you clicked to get to this page. Otherwise enter a date and time to match the date and time you want the backup restored to.

Go

After you click Go, the ZMC validates your entry by looking for an existing backup and displays more options as appropriate. These options are described in the sections that follow.

Restore From

If users are restoring from a **Full Backups** and no incremental backups exist, then the backup set just prior to the time entered will be used to restore the data.



When **incremental backups** exist, ZMC for MySQL provides the ability to restore till the specified time. One of the more common reasons for a restore is to roll back the database to the point before a particular event (such as a mistake or malicious activity) damaged the database. In that case, you should use the Database Events viewer to launch the restore, and all of this information will be automatically prefilled on the **Restore What** page.

If the **Restore from** time is not specified, ZRM restores the most recent full backup and also looks at the subsequent incremental backup for transactions to restore to fulfill the user-specified **Restore to** time.

For example: A full backup was completed on Oct 8 at 16:17:29 and the next incremental backup occurred on Oct 8 at 18:00:00. If you specify a **Restore to** time of Oct 8, 16:17:29 and do not specify a "restore from" time, ZRM restores from the full backup dated Oct 8, 16:17:29 and all transactions that are present in the next incremental backup (Oct 8, 18:00:00) that occurred at Oct 8, 16:17:29.

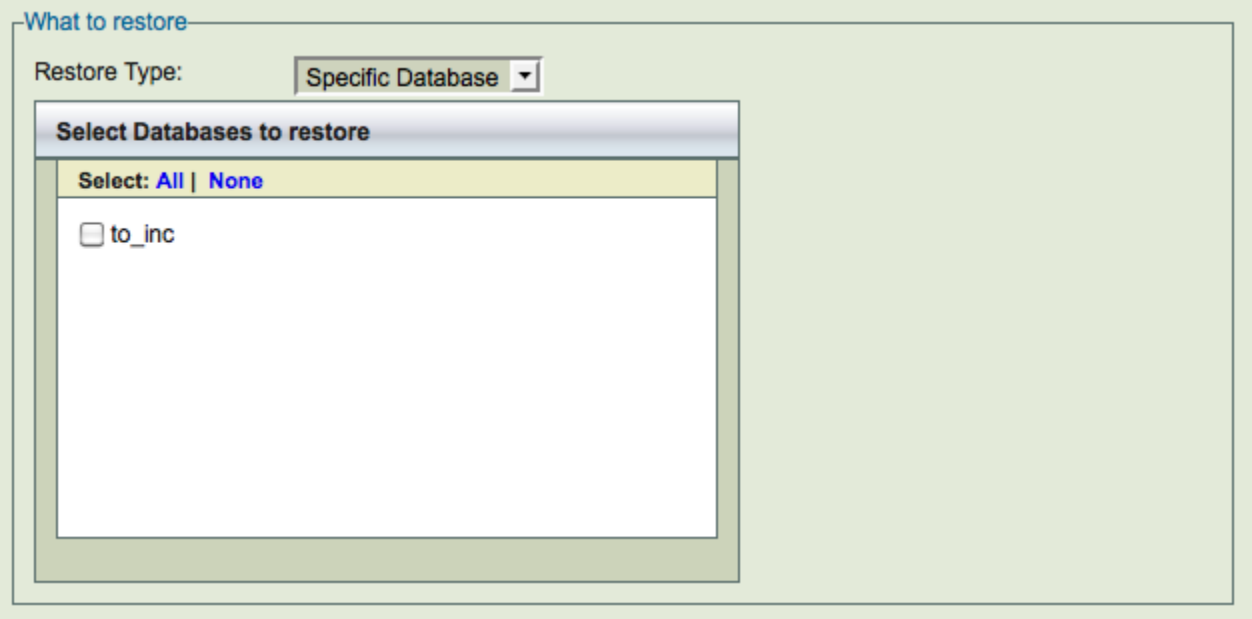
What to Restore

All Databases

When the default choice, **All Databases** is accepted, the GUI does not change any further. Note that in this context the term **All Databases** means all the databases that have been backed up as part of the current backup set; such a restore could be all databases, selected databases, or selected table(s).

Specific Database

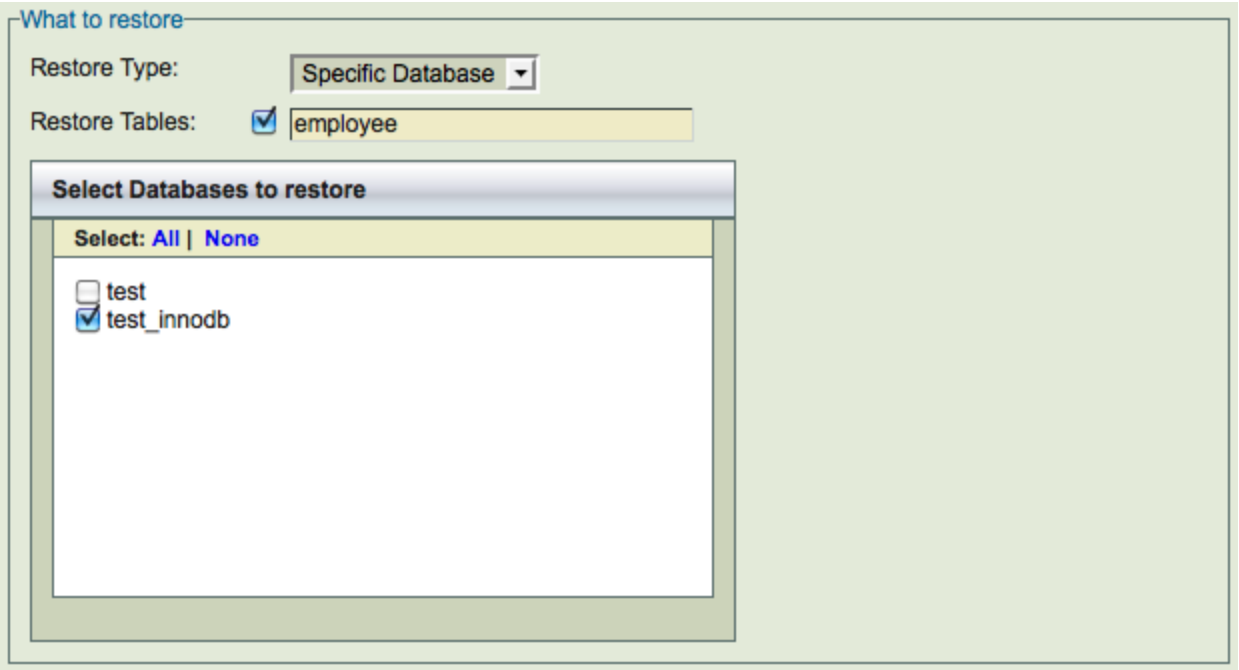
Choose this option to display the list of databases backed up within the set.



Restoration of specific table(s) is not supported with InnoDB storage engine tables. The InnoDB table space contains information about all tables in the database and will have to be restored.

If you have performed backup using Xtrabackup tool, you can restore a table from a database backup to Percona MySQL server. For this functionality, innodb tables should be stored in separate data files i.e, **innodb_file_per_table** must be specified in my.cnf MySQL configuration file. See Percona documentation for the requirements.

Restoration involves the importing the table to the Percona MySQL server and this step has to be performed manually. You can perform the import only to Percona MySQL server.

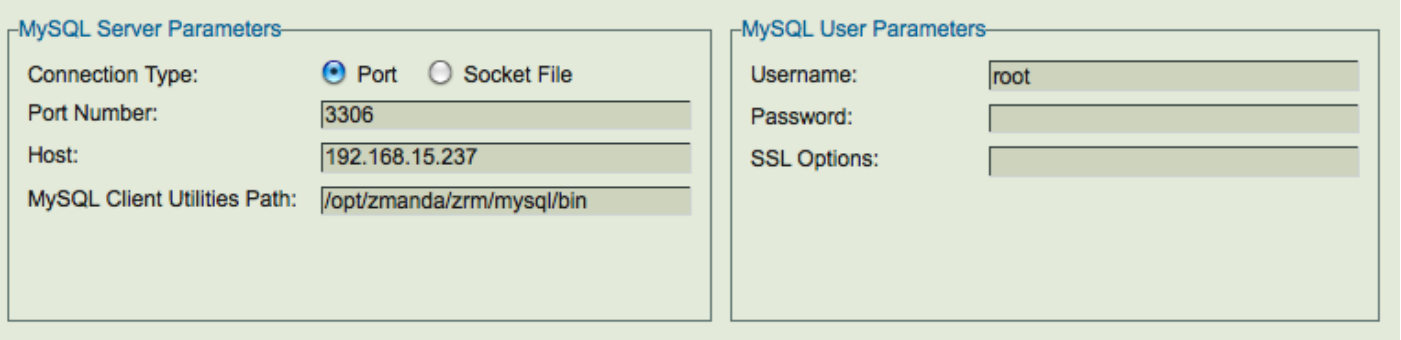


The above figure shows **employee** table from the **test_innodb** database has been selected for restoration. The Restore Tables checkbox and text field will appear only for backups performed using Xtrabackup tool. The names of tables have to be entered in the **Restore Tables** text field.

Clicking Next Step button Initiates an error checking routine, first checking that the Go button has been clicked first. If the Go button has not been clicked, you are prompted to do that. You must correct any errors displayed by clicking Go before you can proceed to the next step. If there are no problems, clicking Next Step takes you to the Restore Where page, with parameters in the Restore What page saved and transferred to Restore Restore page.

Restore Where

The **Restore Where** page lets you specify where the database is to be restored: either the original source, or some other database on some other machine.



These are the same fields on the **Backup What** page MySQL Server Parameters panel describe here. As noted, you can fill them out to connect to the original database (in other words, with the same values used in the **Backup What** page), or fill them out to point to a different server. Click **Next Step** when you are done.

If you are restoring selected databases/tables (especially with InnoDB storage engine), do NOT restore to MySQL server that already has other innoDB tables. MySQL server may not start after the restoration. You should restore a temporary MySQL server or restore the complete backup set that was running on the original MySQL server.

If you are restoring backup images created using MySQL Enterprise Backup tool to a MySQL server, you must have the same data file configuration as the original MySQL server i.e, **innodb_data_file_path** and **innodb_data_home_dir** values must be identical to the original MySQL server.

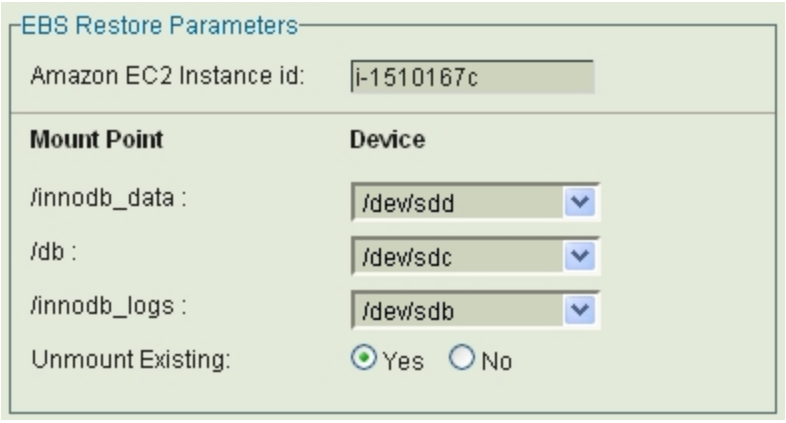
EBS snapshot restore parameters

In addition to other restore parameters, restoration from Amazon Elastic Block Storage snapshot full backups requires the target Amazon EC2 instance id to where the snapshots are restored to. ZMC also provides an option to change device mapping from the original backup. **Device mapping** maps how the EBS volumes and mount points on the original Amazon EC2 instance which was backed up. For example: the mapping at the time of backup could be:

```
/innodb_data : /dev/sda
/db : /dev/sdb
/innodb_logs : /dev/sdd
```

It could be changed at the time of restoration on the restore target EC2 instance. ZRM remembers the original mapping as part of the backup image.

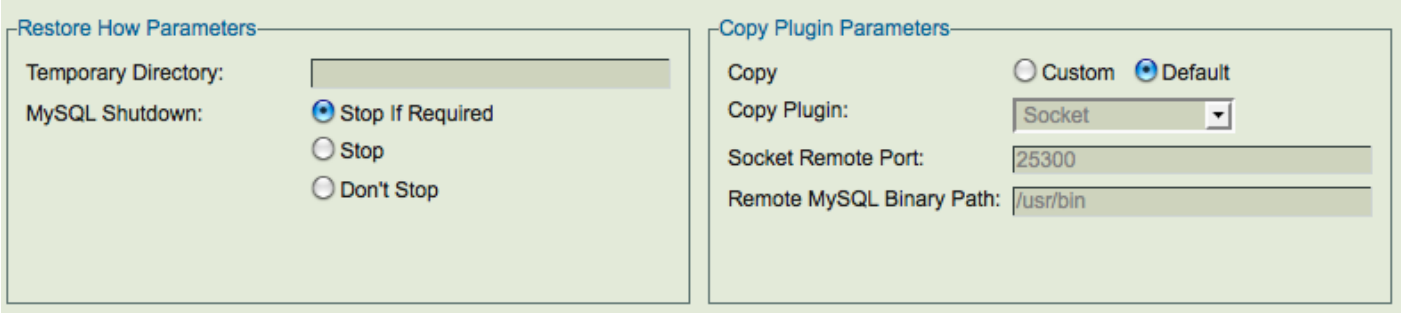
The snapshots are restored and incremental backups are replayed on the target EC2 instance. If there are EBS volumes already mounted at the mount point, the restoration will fail. To prevent this, Check **Unmount Existing** to allow ZRM to unmount existing volumes before restoration of full backups.



Above panel appears only when the backup being restored contains Amazon EBS snapshots.

Restore How

The **Restore How** lets you set some operational details for the restore. If you have set the global settings appropriately in the Site Settings page, you can just use the default values for the controls on this page.



Temporary Directory

ZMC creates temporary files and directories during restore. This directory should have sufficient space to hold at least one full backup on the backup host during restore. By default it will use the directory pointed to by the TMPDIR environment variable. If that is not specified, then **/tmp** is used. Only specify a different directory if you have a specific need to do so.

MySQL Shutdown Options

The 'MySQL Shutdown' radio buttons let you specify whether the MySQL server should shut down during the restore process.

The options are:

- Stop if required.
- Stop.
- Don't stop.

Copy Plugin Parameters

If performing a restore to a remote MySQL server, specify the mechanism used to transfer files between the local ZMC server and the remote MySQL server. The recommended location for plugins is the **/usr/share/mysql-zrm/plugins** directory.

Copy Option

Lets you select whether to use a Copy Plugin during the restore process. Set the 'Copy' radio button to **Yes** to opt for the Copy plugin. If the 'No' option of Copy is selected then rest of the input boxes in the panel can be ignored.

SSH

This plugin uses the Secure Shell protocol to transfer files between the MySQL server and the ZRM server. Since SSH is being used, the data transfer is secure. Unless you have set up SSH keys for the MySQL restore user, you will be prompted for a password when restoring from the backup set.

SSH User

SSH user name

Remote MySQL Binary path

Specify the path on the remote MySQL server under either of the two options.

Socket

The socket based plugin is called **socket-copy.pl**. The socket copy plugin requires MySQL ZRM socket server package to be installed on all MySQL servers. The socket-server package installs **xinetd/inetd** socket server service and restarts **xinetd/inetd**.

The socket copy plugin uses the TCP protocol to transfer data between the MySQL server the ZRM server. It uses port 25300 by default on the MySQL server. You can change the port using the field displayed after you select the socket copy plugin.

Socket Remote Port

Enter the port you wish to use.

Remote MySQL Binary Path

Specify the path on the remote MySQL server where the MySQL binary commands are installed.

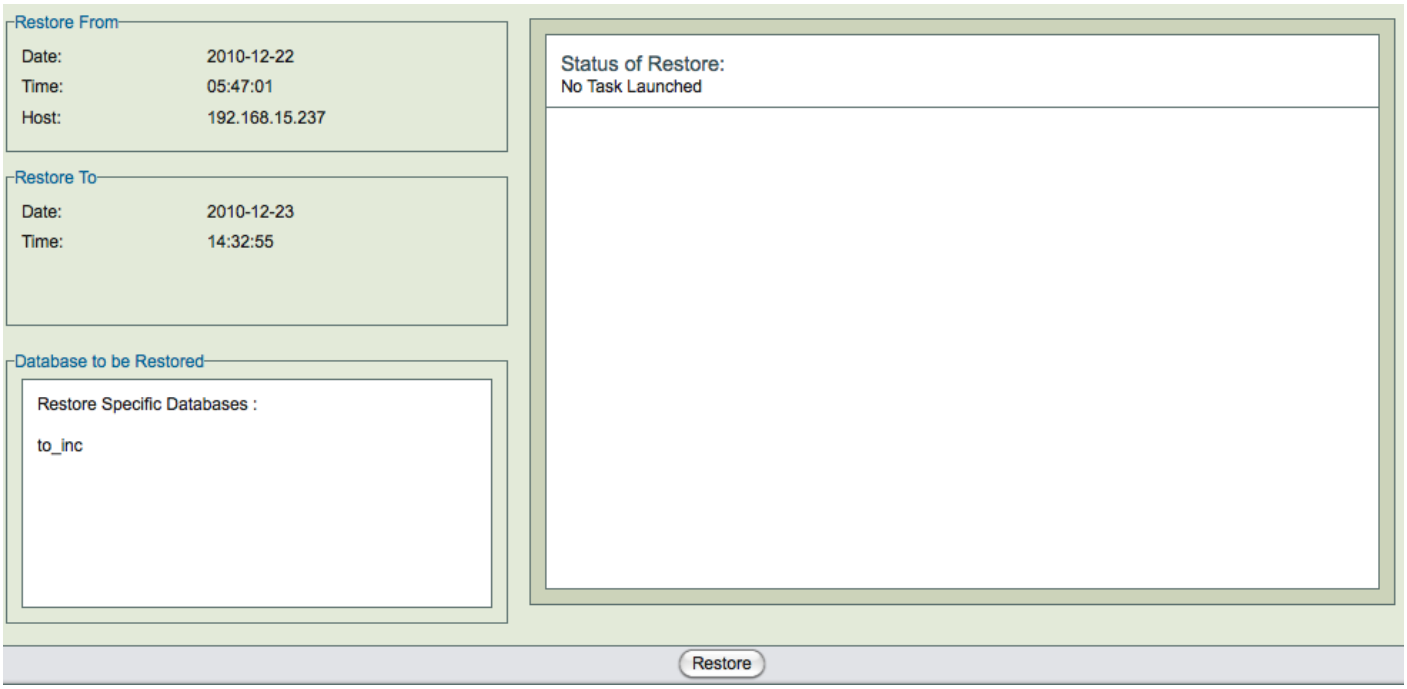
Windows

The Windows copy plugin is required for restoring any Windows-based MySQL server. It requires the ZRM for MySQL Windows client components described in the Installation Instructions. You must then enter the communications ports to use during restore (default is **10081**) operations, and the retry count.

The retry count specifies the number of times ZRM will attempt to restore a raw backup in case shutting down the MySQL server takes more time than expected. Hence ZRM will only attempt to retry for the restore of the very first file that is attempted to be restored. The default value is 2.

Clicking the **Next Step** button at the bottom takes you to the **Run Restore** page.

The **Run Restore** page lets you launch and monitor the restore process that has been selected in the previous dialogs.



The three panels on left side (**Restore From**, **Restore To**, and **Database to be Restored**) of the page contain the data previously gathered. Note the the **Restore To** host is displayed only if it differs from the the **Restore From** host.

The restore process uses disk space under the temporary directory configured in the Backup Where page. The amount of disk space required depends on the number of backup images required for restoration and the backup method used.

Restores can take a long time if the database is large. The status of the restore process is shown in right hand panel. The restore process can be cancelled using the **Cancel** link that appears in the right hand pane. ZRM will cancel the restore process only when it is safe to do so (i.e, will not cause data corruption or loss of data).

Restore task can be cancelled:

1. Before restore process starts
2. When waiting for user input

3. Between restoration of images i.e, i.e, between full backup restoration and incremental backup restoration and between incremental backup restorations.

Only the process running on the ZRM server are stopped during cancellation. All processes on the MySQL server may have to be stopped manually.

After restoration, it is important to check the database(s)/table(s) that were restored. SQL command CHECK TABLE can be used for consistency checking. Use of EXTENDED option is recommended. EXTENDED option does a full key lookup for all rows in the table and will take significant time for a large database.

```
mysql> CHECK TABLE <table1>, <table2> EXTENDED;
```

No other application can be using the database during table consistency check.

Restoring to a MySQL replication slave

This procedure can be used to instantiate a MySQL slave.

You need to follow the procedure from the MySQL manual on how to set up replication. Instead of steps 3 and 5, you can use ZRM backups of the master server to restore the data to the slave server in step 5.

After restoring data to the replication slave either from backup images of MySQL master or another MySQL replication slave, you need to set up configure master server information on the slave.

Perform MySQL CHANGE MASTER TO command on the replication slave as shown below

```
mysql> CHANGE MASTER TO
    ->     MASTER_HOST='replication_master_host_name',
    ->     MASTER_USER='replication_user_name',
    ->     MASTER_PASSWORD='replication_password',
    ->     MASTER_LOG_FILE='value from next-binlog parameter from ZRM reports',
    ->     MASTER_LOG_POS=0;
```

MASTER_LOG_FILE

This value can be obtained from ZRM reports for the backup that was restored. Use ZRM **Custom Reports** page and look at **Next BinLog** parameter for the backup run.

MASTER_LOG_POS

The master-log-position will be zero because logs are rotated when backups are performed.

Restoring a table from a database backup

If you have performed backup using Xtrabackup, you can restore one or more InnoDB table(s) from a database backup. You should selected the tables for restoration in the **Restore Where** page. ZRM restore process exports the innoDB tables.

The exported innoDB tables will have to be manually imported into Percona MySQL server. These steps will have to be performed manually and are documented in the Percona manual.

Notes on Browsers

This section discusses some browser issues you should be aware of when using the Zmanda Management Console.

The Browser "Remember Input" Option

Most browsers have an option to "remember" what is entered in forms. In Firefox, this is set in the **Preferences**->**Privacy** tab. Turning this option on can result in fields being pre-filled with incorrect parameters, as the browser merely remembers the ZMC page and knows nothing about Backup Sets and other ZMC input.

Page Refresh Considerations

When refreshing a page after a POST operation, most browsers display a dialog asking if you want to repeat the operation. Avoid reposting form data when using ZMC. Re-posting form data can result in undesired consequences.

Using the Zmanda Troubleshooting Scripts

Licensed users can collect and send troubleshooting information directly to the Zmanda Support Team using convenient scripts located on ZRM servers and Windows clients.

Running the Support Script on the ZRM server

ZMC includes a support script that gathers various logs helpful for troubleshooting. It includes options for mailing the logs to the Zmanda Support Team (e-mail must be configured and running on the ZRM server).

To start the script, log in as root and run the following commands:

```
# cd /opt/zmanda/zrm/bin
# ./zm-support
```

This generates an archive of all the log files in the current directory. The automatically-generated file name will be displayed at the end of zm-support run:

```
.
.
```

```
.
/opt/zmc/logs/zmc_gui_debug.log
/opt/zmc/logs/zmc_gui_errors.log
/opt/zmc/logs/zmc_installer_16904.log
/opt/zmc/logs/zmc_server.log
/opt/zmc/mysql/data/mysqld.log
/opt/zmc/apache2/logs/access_log
/opt/zmc/apache2/logs/error_log
/opt/zmc/apache2/logs/httpd.pid
/opt/zmc/apache2/logs/NOTEMPTY

Please send this log file -> zm-logs-ZmandaNetworkAccountID-LogID.tar.g
```

You can email the file using your regular e-mail mechanism.

Alternatively, use **zm-support --ship-to-zmanda** to send the logs, assuming e-mail is configured on the Amanda server.

zm-support Options

The **--help** option generates a usage message. Options are detailed below.

--config *ConfigFile1* [, *ConfigFile2*...]
Specifies configuration files to include in the archive. By default, all configuration files are archived.

--skip-config *ConfigFile1* [, *ConfigFile2*...]
Same as above, except this specifies configurations to exclude.

--no-zmc
Excludes ZMC and related information from the archive.

--no-amanda
Excludes Amanda and related information from the archive.

--no-mysql-dump
Excludes the ZMC **mysql** database from the archive.

--no-mysql-zrm
Excludes the ZMC **mysql-zrm** log files from the archive.

--no-var-log-messages
Excludes the **/var/log/messages** file from the archive.

-f or --ftp-to-zmanda
Automatically transfer the archive to Zmanda Support using **ftp**.

-ship
Automatically mails the archive to Zmanda Support.

--no-tar
Rather than creating an archive, create a system information file (**system-info**-*timestamp*) in the current directory, and list the configurations and logs that should be examined or sent to Zmanda Support.

-v
List the files as they are gathered. The default to operate silently.

--help
Display a usage message

Running the Windows Client Support Script

The **zwc-support** utility collects system log files, log files related to ZWC and system related information. The utility then archives these log files into a single compressed file. This compressed file can be then sent to the Zmanda Support team for analysis.

Location

The Zmanda Client for Windows support utility **zwc-support.bat** gets installed when you install the Zmanda Client for Windows. The default location of the utility is **C:\Program Files\Zmanda\Zmanda Client for Windows\bin**.

Files Gathered

The following types of log files are gathered by **zwc-support**:

Zmanda Client for Windows Installation Logs

- C:\amanda_install.log
- C:\amanda_uninstall.log
- C:\Program Files\Zmanda\Zmanda Client for Windows\Debug\LogFile.txt

Zmanda Client for Windows Debug Logs

- C:\Program Files\Zmanda\Zmanda Client for Windows\Debug\LogFile(n).txt
- Zmanda Client for Windows configuration info

System Logs

- System-info
- Application and System Logs

Additional information

- Files and Folders count on all the drives.
- Environment variables list.

Output File

After the utility is run, an output file with the name **zwc-logs-***datetimestamp***.cab** is created in the Zmanda installation directory.

Error Messages

Failure Messages

Permission denied (publickey,keyboard-interactive)

The ZRM server's **mysql** user cannot login to the mysql client using passwordless authentication. You must set up public/private key authentication between the ZRM server and the MySQL server. See this KB article for details.

***mysqlhotcopy_path*:No such file or directory**

The path specified for **mysqlhotcopy** on the **Backup How** page is incorrect. Specify the correct path.

Permission denied at /opt/zmc/mysql/bin/mysqlhotcopy

The uid and gid for the **mysql** user on the ZRM server are different from those on the ZRM server. They must match.

mysqlhotcopy did not succeed

The partition where ZRM is installed is out of space. Free up some space on that partition.

Unable to abort backup run for backup-set

The backup run cannot be aborted due to some delay or error condition. Check for other errors with the backup set (such as snapshot errors) and try again.

mysqldump:Got errno 28 on write

The backup destination directory is full. Free up or add space to the relevant partition and re-run the backup set.

mail command failed

The path to the mail command does not exist. Please check that the ZRM server is correctly configured to send mail. ZRM for MySQL cannot send notifications unless there is a **mail** command installed in one of the following directories: **/usr/local/bin**, **/opt/csw/bin**, **/usr/bin**, **/usr/sbin**, **/sbin:/bin**, or **/usr/ucb**.

Unable to use snapshot
The snapshot-based backup failed for some reason (for example, the volume has exhausted the free extents). Turn on verbose logging and run the backup again to determine the exact cause of the problem. See this KB article for details.

Error backing up
There was a problem with backing up replication files. Check the permissions for the **mysql** user.

Could not copy bin log file
Check that the **bin-log** file exists in the path specified in the **Backup How** page, and that the **mysql** user has the required permissions.

Cannot find bin log file
ZRM for MySQL was unable to back up the MySQL **bin-log** file or replication files. Check that the **bin-log** file exists in the path specified in the **Backup How** page, and that the **mysql** user has the required permissions.

backup failed
This error appears when the backup failed for any of a number of reasons, such as snapshot failure or running out of physical volume extents. Enable verbose logging as described here, then collect the logs and send them to the Zmanda support team as described here.

error opening temporary file
The **mysql** user does not have permission to write to the temporary directory. Set permissions so that the **mysql** user can write to the temporary directory defined on the **Backup Where** page (**/tmp** is the default).

Cannot read tmp file
The **mysql** user does not have permission to read from the temporary directory. Set permissions so that the **mysql** user can read from the temporary directory defined on the **Backup Where** page (**/tmp** is the default).

mysql-zrm appears to be already running
This error occurs when a mysql-zrm process is already running for a backup set. Remove the file **/etc/mysql-zrm/***backup_set_name***/.mysql-zrm.pid**

Could not read file
This error occurs when running in cluster mode, and indicates that the **mysql** user does not have read/write permission on the cluster backup directory. Give the **mysql** user permissions to that directory. Instructions for migrating the **mysql** user from a standard setup to a cluster setup are here.

ndb_mgm
This message indicates that the **MySQL-ndb-tools** and **MySQL-ndb-management** RPM packages are not installed in your system. Install these packages on the target system and try again.

Unable to copy backup dir
The **mysql** backup user does not have permission to read from the cluster backup directory. Ensure that the **mysql** user has read permissions on the cluster backup directory. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

could not delete in directory
The **mysql** backup user does not have permission to write to the backup directory. Ensure that the **mysql** user has write permissions on the backup directory. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

Can not open output file for writing
Ensure that the mysql user has permission to the given directory and file. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

Could not copy database
This error indicates that a copy plugin failed, possibly due to the **mysql** user not having the correct permissions. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

copy-plugin exited with error 512
The copy plugin failed due to any of the reasons listed here, the most likely reason being that the UID or GID of the **mysql** user is different on the client and ZRM server.

Bad retention policy
The values specified for retention policy are not numeric or outside the valid range. Ensure the retention policy is specified as a numeric value, and that it falls within the range of 1 day to 1 year.

Could not copy file
ZRM for MySQL could not copy the bin-log file(s), perhaps because of a permissions problem or a mismatch between the UID/GID of the mysql user on the client and server. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

Error stopping slave
This error occurs if the user lacks permission to stop/start the slave. Make sure that **ssh** keys have been exported properly on all the nodes. Make sure that the **sudoer** file has been configured correctly on the nodes. For more information regarding **sudoer** configuration, see this KB article.

Replication data will not be backed up
This error occurs if MySQL replication is misconfigured or otherwise inoperative, or the Replication peer is unreachable. Make sure that Replication is configured and working properly, and that the Replication peer is reachable.

No write permissions on
This error occurs when the **mysql** backup user does not have write permissions on the destination directory. Ensure that the **mysql** backup user has write permissions on destination directory. For instructions on configuring the correct permissions for the mysql backup user, see this KB article.

Unable to connect to cluster

This message indicates that ZRM is unable to connect to the cluster. There are many issues that can prevent connection to the cluster. Please see the MySQL web site and documentation for further details on cluster errors.

Could not get md5 checksum

This error indicates that the backup has failed the MD5 checksum because it has been corrupted. Take a new backup.

checksum for file does not match

This error indicates that the backup has failed the MD5 checksum because it has been corrupted. Take a new backup.

Errors found during configuration check

The backup set failed the configuration check, perhaps because of a permissions problem or a mismatch between the UID/GID of the **mysql** user on the client and server. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

mysqlbinlog failed for file

There can be many reasons for a binary log restore to fail. Turn on verbose logging and run the restore again to determine the exact cause of the problem. Use the zm-support tool to collect the logs and send them to the Zmanda Support team if you need further assistance.

Backup root directory not found

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

Backup Set directory not found

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

can not parse this index file

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

Can not open index file

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

Restore failed

There can be many reasons for a restore to fail. Turn on verbose logging and run the restore again to determine the exact cause of the problem. Use the zm-support tool to collect the logs and send them to the Zmanda Support team if you need further assistance.

Restore from logical backup failed

There can be many reasons for a logical restore to fail. Turn on verbose logging and run the restore again to determine the exact cause of the problem. Use the zm-support tool to collect the logs and send them to the Zmanda Support team if you need further assistance.

Failed to restore database from raw backup

There can be many reasons for a raw restore to fail. Turn on verbose logging and run the restore again to determine the exact cause of the problem. Use the zm-support tool to collect the logs and send them to the Zmanda Support team if you need further assistance.

Failed to restore innodb databases

There can be many reasons for an InnoDB restore to fail. Turn on verbose logging and run the restore again to determine the exact cause of the problem. Use the zm-support tool to collect the logs and send them to the Zmanda Support team if you need further assistance.

ndb_restore

This message indicates that the **MySQL-ndb-tools** and **MySQL-ndb-management** RPM packages are not installed in your system. Install these packages on the target system and try again.

Restart of cluster node(s) failed

The Cluster failed to restart automatically after a successful restore. Restart the cluster and all nodes manually.

Timeout hit while waiting for nodes(s) to restart

The Cluster failed to restart automatically after a successful restore. Restart the cluster and all nodes manually.

not found in index

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

Errors found during verification
This message indicates a problem with the backup resulting in data corruption. Perform the backup again.

Duplicate entry
MySQL for ZRM has encountered a MySQL bug.

Can not open license file
Indicates that the ZRM for MySQL license file (**/etc/zmanda/zmanda_license**) does not exist, or the **mysql** user does not have permission to read it.

innobackup.pl: No such file or directory
There is a problem with the location and/or execute permissions of **ibbackup/innobackup.pl**. Download the **innobackup.pl** script from www.innodb.com (version 1.5.0 or higher) and install it in the same place that **ibbackup** is installed (usually **/usr/bin**) on the MySQL Server. Make sure the **mysql** user has execute permissions on the file.

sh: ibbackup: command not found
There is a problem with the location and/or execute permissions of **ibbackup/innobackup.pl**. Download the **innobackup.pl** script from www.innodb.com (version 1.5.0 or higher) and install it in the same place that **ibbackup** is installed (usually **/usr/bin**) on the MySQL Server. Make sure the **mysql** user has execute permissions on the file.

Warning Messages

Could not open file
The **mysql** backup user does not have the necessary read/write permission on the backup directory. Ensure that the **mysql** user has write permissions on the backup directory. For instructions on migrating the required permissions to the **mysql** user, see this KB article.

last backup directory is not valid
This message occurs when the last full backup does not exist, either because it was purged or never taken. Perform a full backup as soon as possible.

Error starting slave
This error occurs if the user lacks permission to stop/start the slave. Make sure that **ssh** keys have been exported properly on all the nodes. Make sure that the **sudoer** file has been configured correctly on the nodes. For more information regarding **sudoer** configuration, see this KB article.

Binary logging is OFF

This message indicates that MySQL binary logging is disabled on the MySQL server (backup client). Binary logging is required for incremental backups. Follow these steps to correct the problem:

1. Edit the MySQL configuration file (typically **/etc/my.cnf**) to include the following entry in the **[mysqld]** section:

```
log-bin
```

2. Restart MySQL.

See the MySQL documentation for details on enabling binary logs.

'databases' specified will be ignored

If you choose both **Database** and **Tables** from the same database in a backup set, then the selected**Tables** are backed up and the **Database** selection is ignored.

Database is empty and hence will not be backedup

ZRM for MySQL will not perform a backup on a database that contains no data.

Error getting device details

This message indicates that the **mysql** backup user lacks the necessary **sudo** privileges. For details on **sudoer** configuration see this KB article.

Could not copy table

This message indicates that the **mysql** backup user lacks the necessary **sudo** privileges. For details on **sudoer** configuration see this KB article.

Snapshot create failed

The volume does not have enough free extents available to create the snapshot. Check that you have sufficient free extents available on the volume group to create snapshots. Use **lvdisplay** to check the volume. If you have insufficient free extents then you must add physical volumes to the group. For more information on the procedure to resize an LVM2 logical volume see this Redhat Knowledgebase article.

Snapshot failed as copy failed

The volume does not have enough free extents available to create the snapshot. Check that you have sufficient free extents available on the volume group to create snapshots. Use **lvdisplay** to check the volume. If you have insufficient free extents then you must add physical volumes to the group. For more information on the procedure to resize an LVM2 logical volume see this Redhat Knowledgebase article.

flush tables with read lock and create snapshot failed

The volume does not have enough free extents available to create the snapshot. Check that you have sufficient free extents available on the volume group to create snapshots. Use **lvdisplay** to check the volume. If you have insufficient free extents then you must add physical volumes to the group. For

more information on the procedure to resize an LVM2 logical volume see this Redhat Knowledgebase article.

Unable to open directory

This message indicates that the **mysql** backup user lacks the necessary **sudo** privileges. For details on **sudoer** configuration see this KB article.

umount failed

This message indicates that the **mysql** backup user lacks the necessary **sudo** privileges. For details on **sudoer** configuration see this KB article.

The table(s) of the database will be backed up in logical mode since some of the tables use a transactional engine

To backup a mixed set of databases (non-transactional and transactional), you must either do a logical backup or use snapshots.

It appears that node id is not in started state

This message indicates that the cluster did not automatically restart after a successful restore. Manually restart the cluster and all nodes.

Nothing to do

This message indicates you are trying to restore or verify an unsuccessful backup.

Backup root directory not specified, Assigning default value

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

Backup info fields not specified

This error indicates either of the following:

- An invalid/non-existent destination directory has been specified in the backup set.
- The **mysql** user lacks necessary permissions on the directory.

Note: The destination directory refers to the directory path specified for the backup set in the **Backup Where** page. By default it is /var/lib/mysql-zrm. Make sure that destination directories are present and owned by **mysql** user and group.

Restoring from a backup which was not successful

This message indicates you are trying to restore or verify an unsuccessful backup.

Backup status is not available in the index file

This message indicates you are trying to restore or verify an unsuccessful backup.

Restoring from this may not be successful
This message indicates you are trying to restore or verify an unsuccessful backup.

Restoring from a raw backup without shutting down the MySQL server can result in unexpected problems
You must shut down the MySQL server before performing a raw backup to ensure the database is backed up in a consistent state.

After restore is completed, please verify the restored data
This message is reminder to verify the restored data after a successful restore to ensure that the correct data was in fact restored.

Nothing to verify
This message indicates you are trying to restore or verify an unsuccessful backup.

mysqladmin: command not found
ZRM for MySQL can not find the MySQL executables in the default location, or in the location specified in the **Backup What** page.

/usr/bin/mysql: not found
ZRM for MySQL can not find the MySQL executables in the default location, or in the location specified in the **Backup What** page.

only root can do that
This message indicates that the **mysql** backup user lacks the necessary **sudo** privileges. For details on **sudoer** configuration see this KB article.

Insufficient free extents
The volume does not have enough free extents available to create the snapshot. Check that you have sufficient free extents available on the volume group to create snapshots. Use **lvdisplay** to check the volume. If you have insufficient free extents then you must add physical volumes to the group. For more information on the procedure to resize an LVM2 logical volume see this Redhat Knowledgebase article.